
MARZO 1986
Nº 12 ₳ 2
REP. ARGENTINA
K64
COMPUTACION PARA TODOS
La Odisea de la
Quinta Generación
13 Programas
Inéditos
Concurso:
El Programador
del Año
Suplemento de
Aplicaciones para C64,
CZ 1000/2000, TK 83/90X:
Alarma Anti-robo
Interfase de Control
Procesador de Textos
Sintetizador de Voz



Las puertas del futuro están abiertas en Rosario Usted y una computadora, un nuevo concepto creado por Computational-3 que acercó definitivamente la computadora al hombre Un lugar donde puede operar directamente con las computadoras, jugar con sus secretos, crear nuevos programas Allí encontrará a su disposición todos los modelos de Spectrum, un variado surtido de cassettes para juegos, cassettes y manuales de logo en castellano, Soundbox, y las últimas novedades en materia de computacion Con todo el asesoramiento técnico y la cordialidad de la gente de Computational-3

VENTAJA COMPUTATIONAL-3 PARA LOS LECTORES DE K-64

10% DESCUENTO

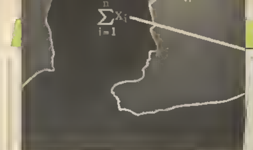
PRESENTANDO ESTE CUPON OBTENDRA UN DESCUENTO DEL 10% PARA SUB-PRODUCTOS.

Usted y una computadora abrio desde Rosario las puertas del mundo del futuro Un mundo donde usted es el protagonista

Computational-3

Ud y una Computadora

EN ROSARIO: Barón de Mauá 1088
C P 2000 Tel 210147
EN RESISTENCIA (CHACO):
Salta 573 - C P 3500 - Tel 28022



SUMARIO

CARTA DEL DIRECTOR

Las profecías de los autores de ciencia ficción vuelven a hacerse realidad, superando incluso lo imaginado. Cuando hace dos décadas Arthur Clark en su "2001: una odisea espacial" describía a la supercomputadora HAL 9000, seguramente no pensaba que las máquinas inteligentes estarían listas diez años antes de la fecha que él había soñado. Y éso es lo que sucederá, con el avance de la quinta generación, de la cual hablamos en este número.

Hoy ya se sabe que la informática no sólo sirve para manejar vuelos en el cosmos, sino que también los pequeños equipos tienen múltiples aplicaciones, y algunos ejemplos damos en nuestro suplemento especial en esta edición.

De manera que las máquinas están al servicio de los hombres, y no a la inversa (como alertó el escritor norteamericano con su HAL 9000)

CRISTIAN PUSSO

QUINTA GENERACION

Los japoneses esperan desarrollar máquinas que piensen como los hombres. La polémica que se generó es analizada por Edward Feigenbaum y Pamela McCorduck.

PAG. 8

ALARMA ANTI-ROBO

Ofrecemos un proyecto con un pequeño programa, para utilizar la Drean Commodore 64 como una alarma, con diversas aplicaciones. Para su instrumentación utilizaremos el USER PORT, el cual es fácilmente accesible en la parte posterior de la C-64

PAG. 27

INTERFASE DE CONTROL

Las máquinas diseñadas en base al microprocesador Z-80 disponen de buenos controladores de periféricos. Uno de ellos, el Z-80 PIO puede ser utilizado para el desarrollo que se explica en estas páginas.

PAG. 34

SINTETIZADOR DE VOZ

Incluimos programas que permiten que nuestra TS 2068, Spectrum o TK 90X "hable" con un vocabulario ilimitado, eliminando las dificultades que puede acarrear el almacenamiento de sonidos.

PAG. 37

PROGRAMAS INEDITOS

TS 1000/1500, CZ 1000/1500, TK 83/85
- Rally 2025 (pág. 18)
- Reloj digital (pág. 20)
- Renumerador (pág. 20)
- Constructor (pág. 20)
- Submarino II (pág. 20)
- Procesador de texto (pág. 30)

SPECTRUM, TS 2068 y TK 90X
- Sintetizador de voz (pág. 37)

TI99/4A
- Demostración de Bit-Map (pág. 44)
- Ruta solar (pág. 46)

Drean Commodore 64
- Procesa-texto (pág. 32)
- Criptografía (Pág. 52)

MSX
- Bloques (pág. 58)

AÑO 1 Nº 12 MARZO DE 1986

K64
COMPUTACION

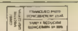
Director General
Ernesto del Castillo
Director Editorial
Cristian Pusso
Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán
Coordinador
M.G. Verdomer Weiss
Secretaria
Moni Ocampo

Diagramación
Fernando Amenguel
Departamento de Avisos
Oscar Devoto

Departamento de Publicidad
Jale Dolores Urien
Promotora
Mónica Garibaldi

K-64 es una Revista mensual editada por Editorial ROEDI S.A. (e/f.), Cerrito 1320, 1º Piso Buenos Aires, Te.: 42-9481/9. Registro Nacional de la Propiedad Intelectual 313.837 M registrado. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.
Impresión: Calcotam. Cromo tapa: Columela. Fotocomposición: Van Waveren
Distribuidor en Capital: MARTINO Juan de Garay 358 Capital T.E. 361 6962. Distribuidor Interior: DGP Hipólito Yrigoyen 1450, Capital T.E. 38 9266/9800. Los ejemplares atrasados se venderán al precio del último número en circulación.



Empresario fracés en Argentina

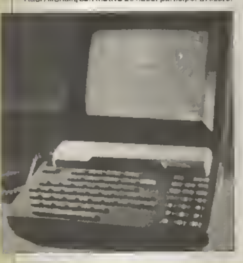
Georges Pebereau

Visitó nnestro país Georges Pebereau, director general del grupo CGE, el primer holding empresario de Francia en el sector electrónico y de telecomunicaciones, que a su vez se ubica dentro de les primeras cinco grandes en el mundo dentro de la especialidad. Entre sus principales áreas se destacan las telecomunicaciones, la informática, la telemática, robotica y el desarrollo de la inteligencia artificial.

Dentro de la actividad que George Pebereau desarrolló en Argentina, se entrevistó con el presidente Raúl Alfonsín, con motivo de hacer participar a Alcatel Thomson, una empresa del holding francés que él dirige, en el concurso para la fabricación de equipos de computación y transmisión numérica. La propuesta presentada por Alcatel Thomson apunta a 120 mil líneas por año, que requerirán una inversión de cien millones de francos.

En cuanto a los componentes, se procurará establecer una relación de cooperación en el campo de la microelectrónica con los laboratorios y las industrias argentinas.

Cabe destacar que Alcatel Thomson es líder mundial en conmutación pública, y, las instalaciones que hasta hoy realizó representen un 30 por ciento del mercado mundial.



Actividad de la Subsecretaría de Informática y Desarrollo

El doctor Carlos María Correa, subsecretario de Informática, dio a conocer que "se han iniciado estudios sobre las nuevas tendencias legislativas en materia de semiconductores de alta integración (chips)". Mencionando el reciente dictado de leyes que establecen un derecho especial de propiedad en EEUU y Japón, señaló que "los países que buscan ingresar al campo de la informática deben estudiar el impacto de estas legislaciones, así como el convenio internacional sobre la materia qne ya ha sido elaborado y propuesto por la Organización Mnndial de la Propiedad Intelectual" Adelantó también que "un representante de nuestro país asistirá el próximo mes a nna rennión internacional sobre el tema, y procurará sentar posiciones comunes con otros países en desarrollo, y en particular los latinoamericanos".

La informática y los derechos indivuales

Por otro lado, el doctor Correa comentó el avance registrado en los trabajos de la misión que analiza el impacto de la Informática sobre la privacidad y los derechos individuales. "La manipulación de datos de las personas —recordó— pone en riesgo no sólo la confidencialidad de información personal, sino que genere indefensión frente a datos cuya exactitud no se puede corroborar y corregir". Dentro del trabajo realizado, el funcionario comentó que la comisión "realizó un análisis de la legislación comparada, en

particular de las leyes austríaca, danesa, francesa y noruega, e identificó doce principios generales".
Algunos de los principios son, de la justificación social; de la limitación de la recolección; de la calidad de la información; de la especificación del propósito o la finalidad; de la confiabilidad; de salvaguarda de la seguridad, de política o apertura, de la limitación en el tiempo; de control y de la participación individual.
Esta comisión que lleva adelante estos estudios está integrada por los doctores Carlos Nino, Ricardo Guibourg, Pedro Molinero y Eduardo Novoa Monreal.

Publicación de NCR Argentina

La empresa NCR ha lanzado el primer número de Presencia, publicación institucional de la empresa en Argentina. Esta nueva publicación, que se editará trimestralmente, se distribuye a consultores y a clientes actuales y potenciales. En ella se encontrarán testimonios de usuarios de equipos NCR, información sobre sistemas que se presenten en el mercado y artículos sobre novedades de la empresa.

Combaten la piratería

Dos nuevos recursos para combatir la piratería de software están prontos a ser utilizados. El primero de ellos consiste en una lente especial que se suministra junto con el soft, con el cual hay que leer un código que aparece en la pantalla. Este sistema permite realizar copias "back-up" de seguridad para el propietario del soft.
El otro sistema resultará útil también para los comercios expendedores de soft. Consiste en programas en "cartridges" del tamaño de una tarjeta de crédito que se graban en el momento de venderse. Para usarlos, se necesitará de una interface especial de bajo costo, adaptable a las marcas más populares.
Entre otras ventajas, los programmas se cargarán inmediatamente, ocuparán poco espacio físico y no correrán peligro de borrarse.

Recibieron sus premios

La foto registra el momento en que Jorge Mariaschin y Jorge Rossi concurrieron a nuestras oficinas para retirar los premios que obtuvieron como ganadores del tercer concurso K-64.
El primero recibió los dos pasajes a Bariloche, ida y vuelta en avión, mientras que el segundo se llevó



Jorge Rossi (segundo premio), y Jorge Mariaschin (primer premio).

los dos boletos para Punta del Este, también ida y vuelta en avión. Como se recordará los otros galardonados ganaron una impresora Alphacom 32, un grabador para computadora y una mesa especial

para informática. ¿Quién se adjudicará el lingote de oro de la nueva selección K-64 "El programador del año '86"? (ver bases en esta número).

Sistema/36 PC de IBM

IBM anunció el lanzamiento en el mercado argentino del Sistema/36 PC (IBM 5364) que debido a su gran variedad de aplicaciones, es un equipo apto para servicios profesionales, comerciales y como procesador de datos para la pequeña y mediana empresa.
Es de fácil aplicación y de costo accesible. Su tamaño permite que pueda colocarse sobre un escritorio. En cuanto a sus particularidades técnicas podemos mencionar que soporta un máximo de cuatro estaciones de trabajo conectadas localmente, la primera de las cuales debe ser una PC, y hasta 64 estaciones adicionales remotas conectadas por líneas de comunicación. Por otra parte, utiliza un disco fijo desarrollado y fabricado por IBM con una capacidad de almacenamiento de 40 millones de caracteres, que



Nuevo componente del S/36 PC

se puede ampliar a 80 millones de caracteres en una segunda unidad. Además cuenta con una unidad de diskette que pueda leer y grabar diskettes de 1,2 Megabytes de capacidad.
Una importante ventaja del sistema Sistema/36 PC es la posibilidad de acceder a todo el software que hay disponible en el mercado local para Sistema/36.

Escuela Superior Latinoamericana de Informática

La Escuela Superior Latinoamericana de Informática (ESLAI), creada en 1985 bajo el auspicio del gobierno argentino y de organismos internacionales especializados inició en octubre de ese mismo año sus actividades regulares, dirigidas a estimular la formación de alto nivel de docentes e investigadores latinoamericanos en informática.

COMMODORE 128

C128 en Argentina

Es posible encontrar, en nuestro mercado, la nueva COMMODORE 128 PERSONAL COMPUTER. Esta poderosísima computadora puede definirse como "tres en una" ya que puede trabajar en modo C-64 permitiendo que todos los programas escritos para él puedan correrse en la 128, en el modo C-128 con el nuevo BASIC 7.0 y en modo CP/M trabajando con el microprocesador Z80. En este último modo se puede correr los cientos de programas comerciales disponibles para computadoras de 8 bytes con CP/M como sistema operativo.
Además puede leer ocho formatos de CP/M incluyendo aquellos para OSBORNE, EPSON e IBM CP/M 86.
Commodore lanzó en EE.UU junto con la PC el nuevo Disk Drive 1571 que presenta, también, tres modos de operación. Cuando la C-128 trabaja en modo 64, el 1571 se comporta como una 1541 cien por cien. Respeta la baja velocidad de transferencia de información de ésta (300 cps) y su capacidad de almacenamiento (170 Kbytes). En modo 128, la velocidad de transferencia aumenta a 1500 cps (5 veces más rápido que la 1541) y la capacidad se eleva a 350 Kbytes.
Finalmente, en el modo CPM, la velocidad de transferencia llega a los 3500 cps (de 10 a 12 veces más rápido que la 1541) con cerca de 410K de almacenamiento de datos.
Algunas de sus características son:
Microprocesador:
Z80 (modo CP/M)
Memoria:
128K RAM (expandible a 512K)
Clock:
4 MHz (modo CP/M)
1 o 2 MHz (modo 64)
Almacenamiento en disco (también trabaja en cassette):
350-410K: 128 / CP/M
170K: 64
Sonido:
3 voces con generador de ruidos
Teclado: 92 teclas, 14 numéricas, 8 de función.
Pantalla:
40/80 columnas
Otro detalle importante es que puede trabajar en alta resolución directamente

LA ODISEA DE LA QUINTA GENERACION

En menos de una década los japoneses esperan desarrollar máquinas superdotadas que piensen como los hombres. La promesa desató polémicas, que son analizadas en el libro "La Quinta Generación", de Edward Feigenbaum y Pamela McCorduck, reeditado por Sudamericana-Planeta, del que publicamos algunos comentarios.

Una máquina tan inteligente como una persona

La dificultad con que la mayoría de nosotros se enfrenta cuando ha de pensar en máquinas inteligentes es que nuestro concepto de «máquina"» está condicionado por las máquinas que nos han rodeado toda la vida. Su función en casi todos los casos es tratar la energía: es decir, ampliar, distribuir, transformar o modificar de algún modo la energía. Así, por ejemplo, el automóvil transforma la energía de un combustible fósil (a su vez transformado ya por el refinamiento) en energía cinética, y esta transformación amplifica la energía cinética humana y satisface objetivos humanos. En coche podemos ir más lejos que andando. Además todas estas transformaciones de la energía pueden describirse claramente mediante las disciplinas científicas clásicas.

Sin embargo, el ordenador es un tipo diferente de máquina. No trata energía, sino información. Como es lógico, entra alguna energía en el proceso, del mismo modo que la transformación de la información interviene también en los sistemas telefónicos o de radiodifusión, pero las transformaciones de energía en el ordenador son sus rasgos menos interesantes, aunque puedan serlo para algunos ingenieros. Para comprender la función esencial de los ordenadores —en su calidad de máquinas— debemos quitamos de la cabeza las viejas metáforas y empezar a pensar de un modo nuevo. El ordenador es el aparato principal de la era de la información. Su objetivo es sin duda tratar la información: transformarla, amplificarla, distribuirla y modificarla en general. Pero hay algo más importante, y es que el ordenador produce información. La esencia de la revolución informática es que la carga que supone producir el saber futuro del mundo recaerá sobre los aparatos mecánicos y no sobre las cabezas humanas. Aunque el Eclesiastés diga lo contrario, quizá haya algo nuevo bajo el sol.

'*Debemos estar orgullosos de reconocer nuestras limitaciones y de inventar una tecnología que pueda compensarlas*".

Con todo, estos aparatos no han recibido el nombre adecuado, y esto puede confundirnos. La palabra computadora u ordenador, con sus reminiscencias de cálculos y contaje, sólo nos informa sobre la utilización histórica de la máquina, no sobre su potencial. Los japoneses se han dado cuenta de ello y han bautizado a su Quinta Generación de ordenadores con el nombre de máquinas para «el tratamiento informático del saber», lo cual sugiere a su vez que el saber y la información son dos entidades distintas.

Nosotros hemos vivido una época de transición, con teléfonos y televisores que están a caballo de ambos mundos, el de la información y el de la energía. La llegada de la primera generación de ordenadores nos introdujo claramente en la nueva era. Y ahora estamos avanzando hacia la fase siguiente: la era de las Máquinas Inteligentes.

Sentimos en nuestra nuca el cálido aliento de los maniáticos «¿Qué significa esto de inteligente? Estas máquinas pretendidamente inteligentes, ¿no lo serán tanto como una persona, supongo? Es imposible que lo sean, porque todo lo que saben se lo enseñan las personas.»

«Piensa —dice Feigenbaum a McCorduck un día— que no existe ninguna máquina que sea tan inteligente como una persona.»

Ella le mira sorprendida. ¿Son una estafa todos estos programas que generan la capacidad de los especialistas? ¿Quizás ella no ha captado bien sus palabras, y le pide que lo repita, pero tampoco así acaba de entenderlo.

«¿Podrías explicármelo mejor?.»

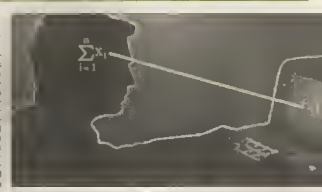
«Muy fácilmente. se empieza escogiendo una tarea para que una máquina la lleve a cabo. Se detalla esta tarea muy precisamente, basándose en el saber humano. Se utilizan para ello los conocimientos del equipo de especialistas, pero la máquina continúa siendo menos inteligente que ellos. Sin embargo, el hecho de tener el programa y los conocimientos expuestos detalladamente permite descubrir de modo inmediato la manera de introducir mejoras. Y de repente el programa supera al hombre. Resulta imposible fijar el momento preciso en que la máquina es exactamente tan inteligente como una persona. Durante un tiempo no era tan inteligente y de pronto se convierte en más inteligente.

«Las máquinas cuidan metódicamente del detalle, son incapaces de cansarse, son inmunes al aburrimiento y tienen una elevadísima velocidad. Todo ello, unido a su poder de raciocinio y a su información, está empezando a producir

un saber que a menudo es más rápido y mejor —más «inteligente»— que el de las personas que los enseñaron.

»Y debemos preguntarnos con toda humildad: ¿Qué inteligentes eran las personas que enseñaron a estas máquinas? Dentro de la escala temporal de la evolución, los animales pensantes son unos recién llegados. La evolución no dispuso de mucho tiempo para dedicarse a perfeccionar la capacidad de conocimiento del hombre. Sin duda las respuestas correctas a las preguntas: "¿Qué complejo patológico está afectando a mi paciente?" o "¿Qué plan experimental permitirá obtener un clon de un gen particular?", o "¿Cómo puedo sintetizar el nuevo fármaco que acabo de descubrir?" están debajo de nuestras mismas narices, pero somos incapaces de verlas. Sin embargo existen ya programas de sistemas especializados, claramente primitivos, que pueden responder a tales preguntas. En el futuro, máquinas más inteligentes responderán de modo rutinario a preguntas más difíciles.

»Las personas sabemos muy bien convertir las señales sensoriales en símbolos de conocimiento y resolver problemas que exigen sentido común. Pero cuando nos enfrentamos con grandes cantidades de datos nos acobardamos: nos mostramos poco sistemáticos y olvidadizos, nos aburrimos, nos distraemos. La escritura y la tecnología del libro nos ayudaron a superar algunos de estos problemas, los ordenadores inteligentes e interactivos nos ayudarán algo más. Debemos sentirnos orgullosos de poder reconocer nuestras limitaciones y de inventar una tecnología que puede compensarlas.»



NOTA. IA (Inteligencia Artificial)

La fe en la inteligencia artificial

Un participante en la conferencia de la Quinta Generación se levantó para formular algunas objeciones a lo que había oído. Las objeciones no eran serias, pero el participante concluyó así: «Creo, resumiendo mi punto de vista, que estamos interesados en la siguiente generación de ordenadores, y que de momento suponemos que estos ordenadores serán máquinas de IA. En términos generales estoy de acuerdo con ustedes, pero no quisiera perder de vista que algunas personas asistentes no creen en la IA y que, por tanto, desearían considerar a los ordenadores de la Quinta Generación como algo diferente.» La expresión era curiosa: «no creen en la IA», como si la IA fuera cuestión de fe mística, como si fuera algo no sujeto a prueba empírica. De hecho la expresión eludía a una controversia de trascendencia mucho mayor que la referente a la elección del mejor lenguaje de programación posible o a la idoneidad del enfoque basado en el saber para conseguir que los ordenadores se comporten de modo inteligente, o a cualquier otra de las disputas científicas que han animado el campo de la investigación de la inteligencia artificial en su más de un cuarto de siglo de existencia.

«... el saber futuro del ...»
«... era sobre los aparatos ...»
«... no sobre las cabezas hu...»

Decidir si la Quinta Generación sería una máquina de razonamiento simbólico con un diseño nuevo o más bien una versión mayor y mejor de las cuatro primeras generaciones de ordenadores era algo que se resolvería a su debido tiempo.

Lo que no se resolvería, por lo menos en las mentes de quienes lo dudaban de entrada, era la posibilidad de creer en la inteligencia artificial. Decir que uno no cree en la inteligencia artificial —y hay mu-

EL ARTE DE CONSTRUIR UN CEREBRO

Hace menos de dos décadas, Arthur C. Clarke, en "2001, una odisea espacial" (reeditado ahora por Hyspamérico), anticipaba el nacimiento de las extrañas criaturas tecnológicas.

Al sexto miembro de la tripulación no le importaban nada todas esas cosas, pues no era humano.

Era el sumamente perfeccionado computador HAL 9.000, cerebro y sistema nervioso de la nave.

HAL (sigla de Computador ALgorítmico Heurísticamente programado, nada menos) era una obra maestra de la tercera promoción de computadoras. Ella parecía ocurrir a intervalos de veinte años, y mucha gente pensaba ya que otra nueva creación era inminente.

La primera había acontecido en 1940 y pico, cuando la válvula de vacío hacía tiempo anticuada, había hecho posible tan toscos cachivaches de alta velocidad como ENIAC y sus sucesoras. Luego, en los años sesenta habían sido perfeccionados sólidos ingenios microelectrónicos. Con su advenimiento, resultaba claro que inteligencias artificiales cuando menos tan poderosas como la del Hombre, no necesitaban ser mayores por mesas de despacho... caso de que se supiera cómo construirlas.

Probablemente nadie lo sabría nunca; mas ello no importaba. En los años ochenta, Minsky y Good habían mostrado cómo podían ser generadas automáticamente redes nerviosas autorreplicadas, de acuerdo con cualquier arbitrario programa de enseñanza. Podían construirse cerebros artificiales mediante un proceso asombrosamente análogo al desarrollo de un cerebro humano. En cualquier caso dado, jamás se sabrían los detalles precisos, y hasta si lo fueran, serían millones de veces demasiado complejos para la comprensión humana.

Sea como fuera, el resultado final fue una máquina-inteligencia que podía reproducir —algunos filósofos preferían aún emplear la palabra "remedar"— la mayoría de las actividades del cerebro humano, y con mucha mayor velocidad y seguridad. Era sumamente costosa, y sólo habían sido construidas hasta la fecha unas cuantas unidades de la HAL 9.000; pero estaba ya comenzando a sonar un tanto a hueca la vieja chanza de que siempre sería más fácil hacer cerebros orgánicos mediante un inhábil trabajo.

Hal había sido entrenado para aquella misión tan esmeradamente como sus colegas humanos... y a un grado de potencia mucho mayor, pues además de su velocidad intrínseca, no dormía nunca. Su primera tarea era mantener en su punto los sistemas de subsistencia, comprobando continuamente la presión del oxígeno, la temperatura, el ajuste del casco, la radiación y todos los demás factores inherentes de los que dependían las vidas del frágil cargamento humano. Podía efectuar las intrincadas correcciones de navegación, y ejecutar las necesarias maniobras de vuelo cuando era el momento de cambiar de rumbo. Y podía atender a los hibernadores, verificando cualquier ajuste necesario a su ambiente, y distribuyendo las minúsculas cantidades de fluidos intravenosos que los mantenían con vida.

Las primeras generaciones de computadoras habían recibido la fuerza necesaria a través de teclados de máquinas de escribir aumentadas, y

chas personas que lo afirman añadiendo todo tipo de énfasis, razones e irritación— equivale a decir que uno no cree que de una máquina pueda decirse que piense, haga lo que haga esta máquina.

Cuando alguien propuso por primera vez que quizás un ordenador llegaría a comportarse inteligentemente, se levantó un enérgico coro de protestas. Por inteligente que fuera el comportamiento de los sucesivos ordenadores, los incrédulos no se dejarían convencer. La misma frase de «creer en» sugiere el dogma, la doctrina, o el poder o no poder según los catecismos. *«Moi, je suis socialiste»*, dice e Stephen Daedalus uno de sus amigos. *«Je ne crois pas en l'existence de Dieu.»* Por mi parte yo trabajo con componentes y circuitos. Yo no creo en la existencia de la inteligencia artificial.

Feigenbaum había escuchado este argumento tan a menudo que ahora podía contar una pequeña historia sobre el tema. La protagonizaba el gran físico Niels Bohr, a quien visitaba uno de los jóvenes físicos de Europa. El joven científico se escandalizó cuando vio una herradura clavada sobre la puerta del gran físico. «Profesor Bohr —le dijo—, ¿supongo que usted no cree en esta vieja superstición?» El físico reflexionó un momento y luego respondió alegremente: «Dicen que surte efecto tanto si uno cree en ella como si no.»

Una escafandra para la mente

Una de las objeciones que los científicos vecinos formulan contra la inteligencia artificial son las profecías, en su opinión disparatadas e incluso irresponsables, hechas por quienes trabajan en este campo.

> *"Máquinas más inteligentes responderán de modo rutinario a preguntas cada vez mas difíciles".*

habían replicado a través de impresoras de alta velocidad y despliegues visuales. Hal podía hacerlo también así, de ser necesario, pero la mayoría de sus comunicaciones con sus camaradas de navegación se hacían mediante la palabra hablada. Poole y Bowman podían hablar a Hal como si fuese un ser humano, y él replicaría en el perfecto y más puro inglés que había aprendido durante las fugaces semanas de su electrónica infancia.

Sobre si Hal pudiera realmente pensar, era una cuestión que había sido establecida por el matemático inglés Alan Turing en los años cuarenta. Turing había señalado que, si se podía llevar a cabo una prolongada conversación con una máquina —indistintamente mediante máquina de escribir o micrófono— sin ser capaz de distinguir entre sus respuestas y las que pudiera dar un hombre, en tal caso la máquina estaba pensando, por cualquier sensible definición de la palabra. Hal podía pasar con facilidad el test de Turing.

Y hasta podía llegar el día en que Hal tomase el mando de la nave. En caso de emergencia, si nadie respondía a sus señales, intentaría despertar a los durmientes miembros de la tripulación, mediante una estimulación eléctrica y química. Y si no respondían, pediría nuevas órdenes por radio a la Tierra.

Y entonces, si tampoco la Tierra respondiese, adoptaría las medidas que juzgara necesarias para salvaguardia de la nave y la continuación de la misión... cuyo real propósito sólo él conocía, y que sus colegas humanos jamás habrían sospechado.

Poole y Bowman se habían referido a menudo humorísticamente a sí mismos como celadores o conserjes a bordo de una nave que podía realmente andar por sí misma. Se hubiesen asombrado mucho, y su indignación hubiera sido más que regular, al descubrir cuánta verdad contenía su chanza.

....

Había otros pensadores —Bowman lo hallaba así también— que sustentaban puntos de vista aún más avanzados. No creían que seres realmente evolucionados poseyeran en absoluto un cuerpo orgánico. Más pronto o más tarde, al progresar su conocimiento científico, se desembarazarían de la morada, propensa a las dolencias y a los accidentes, que la Naturaleza les había dado, y que los condenaban a una muerte inevitable. Reemplazarían su cuerpo natural a medida que se desgastasen —o quizás antes— con construcciones de metal o de plástico, logrando así la inmortalidad. El cerebro podría demorarse algo como último resto del cuerpo orgánico, dirigiendo sus miembros mecánicos y observando el Universo a través de sus sentidos electrónicos... sentidos mucho más finos y sutiles que aquellos que la ciega evolución pudiera desarrollar jamás.

Hasta en la Tierra se habían dado ya los primeros pasos en esa dirección. Había millones de hombres, que en otras épocas hubiesen sido condenados, que ahora vivían activos y felices gracias a miembros artificiales, riñones, pulmones y corazones. A este proceso sólo cabía una conclusión... por muy lejana que pudiera estar.

Y eventualmente, hasta el cerebro podría incluirse en él. No resultaba esencial como sede de la conciencia, como lo había probado el desarrollo de la inteligencia electrónica. El conflicto entre mente y máquina podía ser resuelto al fin en la tregua eterna de la completa simbiosis...

Mas, ¿era aún esto el fin? Unos cuantos biólogos inclinados a la mística, iban todavía más lejos. Atando cabos en las creencias de diversas religiones, especulaban que la mente terminaría por liberarse de la materia. El cuerpo-robot, como el de carne y hueso, sería solamente un peldaño hacia algo que, hacía tiempo, habían llamado los hombres "espíritu".

Y si más allá de esto había algo, su nombre sólo podía ser Dios.

¿FICCION O CIENCIA?

Es cierto que se han hecho profecías que no se han cumplido todavía. Por ejemplo, un conjunto de predicciones en 1958 dijo que al cabo de diez años un ordenador sería el campeón mundial de ajedrez. Pasaron diez años y al cabo de este tiempo la mayoría de los científicos habían perdido interés en el ajedrez de ordenador. Pero pasados veinte años los ordenadores jugaban el ajedrez con una categoría suficiante para ganar en los campeonatos. Casi toda la investigación que había impulsado a estos ordenadores por encima del nivel de principiantes se había hecho con el equivalente en IA de los aficionados caseros. Las máquinas de ajedrez que actualmente funcionan a nivel de campeonato y que, por tanto, juegan el ajedrez mejor que el 99 por ciento de nosotros, se deben todavía al trabajo apasionado de un pequeño grupo de investigadores y no constituyen, como se había pensado al hacer aquella predicción, un campo central de pruebas donde descubrir reglas de comportamiento inteligente. Desde el punto de vista intelectual, un buen ajedrecista no es ni más ni menos que un buen ajedrecista. Como veremos más tarde, este resultado permitió hacer un descubrimiento importente sobre la inteligencia como especialización del saber.

Los especialistas de todos los campos no se cansan nunca de formular predicciones sobre el futuro. Si se hacen comparaciones, la inteligencia artificial se ha acercado mucho más el cumplimiento de sus profecías que muchas otras ramas de la ciencia. Es evidente que las predicciones cumplen funciones importantes de tipo psicológico, social y planificador, sea cual fuere su correspondencia con el futuro eventual. ¿Por qué, entonces, tanta gente se sienta ofendida cuando se hacen predicciones sobre la inteligencia artificial?

La respuesta a esta pregunta parece evidente. El elemento ofensivo en las predicciones formuladas por los investigadores de inteligencia artificial es el mismo elemento que ofende a algunas personas cuando consideran la idea de la inteligencia artificial en sí misma: es el hecho de que exista. Está claro que los científicos están creando ya máquinas inteligentes cuyo objetivo es amplificar la inteligencia humana, una especie de escafandra autónoma, que permitirá a la mente humana ir a lugares que hasta ahora le estaban vedados, e lugares que según algunos quizá deberían estarles vedados. Es evidente que las personas ofendidas no creen que la inteligencia artificial ejerza la función liberadora de una escafandre autónoma. La inteligencia artificial amenaza de modo profundo y no muy sutil al concepto que tienen ellos de sí mismos: Nuestra identidad como persona deriva de nuestra inteligencia. La idea de que haya otras cosas —y peor aún de que haya cosas creadas por nosotros— que también puedan ser inteligentes requiere una revisión considerable de nuestra imagen.

Los científicos están creando máquinas cuya objetiva es amplificar la inteligencia, una especie de escafandra autónama.

Los intelectuales están experimentando de modo muy real e inmediato lo que muchos otros trabajadores han sufrido con anterioridad: la sustitución de sus habilidades especiales por máquinas. El profesor Edward Fredkin, del MIT, ofreció en cierta ocasión una perspectiva sobre el tema: «Los seres humanos no están mal. Estoy contento de ser uno de ellos. En general me gustan, pero sólo son humanos. Tampoco me quejaré. Las personas no son los mejores excavadores de zanjas del mundo, lo son las máquinas. Y las personas no pueden levantar los pesos que levanta una grúa. No pueden ni volar, si no se meten en un avión. Y no pueden transportar lo que transporta un camión. A mí esto no me ofende en absoluto. Hubo personas cuya vida giraba alrededor de algo exclusivamente físico: John Henry y el martillo de vapor. Ahora nos enfrentamos con el martillo de vapor intelectual. Al intelectual no le gusta que esta máquina lo haga mejor que él, y su reacción es idéntica a la del trabajador superado físicamente por la máquina.»

Hay otras personas como Fredkin, cuyas identidades no se sienten amenazadas lo más mínimo por las posibilidades de una máquina inteligente. La trascendencia de un hecho así los emociona. No los preocupa mucho que el fenómeno se está produciendo paso a paso y no de la noche a la mañana; esta lentitud sólo afecta a quienes no sólo desean que el fenómeno se produzca, sino que desearían acelerarlo porque hay muchas cosas por saber y por hacer y una máquina inteligente los ayudaría a saberlas y a hacerlas más rápidamente. También hay entre ellos personas en cuya opinión, el término *inteligencia* está recargado de palabrería seudocientífica y que precisa algo de rigor empírico. En su opinión, asignar la palabra «inteligencia» al comportamiento de un ordenador no supone una apostasía excesiva. Y quizás en esto resida una de las explicaciones de la tranquilidad de McCorduck ante el hecho de la inteligencia artificial. La comprensión del tema no le llegó de repente. Empezó a darse cuenta mientras comparaba los argumentos en contra de las máquinas pensantes con las razones presentadas en el siglo XIX para demostrar que las mujeres no podrían estar nunca el mismo nivel intelectual que los hombres. McCorduck encontró paralelismos divertidos entre ambos argumentos. Al principio parecían únicamente un material jocoso y fácil de criar sobre la imposibilidad de que las mujeres pudiesen pensar realmente —razones de emoción, diferencias insuperables entre hombres y mujeres, la falta de precedentes, y además, sí, consideraciones éticas—, pero paulatinamente empezó a comprender que todo aquello ocultaba una verdad mayor. La *inteligencia* era un término político, definido por quienes mandan en cada momento. Esto explicaba su asombrosa elasticidad. Y entonces la cuestión de si una máquina pueda pensar se convirtió de repente para McCorduck en una no cuestión, en un no tema absolutamente inconsecuente. Sin embargo, se plantea una cuestión más básica, a saber: ¿Importa realmente el ordenador? La respuesta es afirmativa, tanto en un sentido personal como global. El ordenador tiene importancia para el conjunto de nosotros, pero también tiene importancia para cada uno de nosotros. K64

LAS TS 2068 Y SPECTRUM POR DENTRO

Como ya lo habíamos mencionado en nuestro artículo anterior, ambas computadoras poseen el mismo microprocesador Z-80 de Zilog.

Este es un microprocesador de tercera generación. los microprocesadores de Intel 8008 y 8080 fueron sus antecesores inmediatos. La arquitectura interna del Z-80 es similar a la de estos micros. El set de instrucciones del Z-80 posee 158 tipos distintos contra las 58 instrucciones del 8008 y las 78 instrucciones del 8080.

Arquitectura del Z-80

Según podemos observar en la figura 1, donde se representa el diagrama en bloques del Z-80, éste se encuentra organizado en distintos buses, siendo un bus un conjunto de conexiones eléctricas que cumplen una función similar. Por ejemplo, todos los datos que entran y salen del micro pasan por el bus de datos que, en este caso, por ser de ocho conexiones eléctricas, diremos que es un micro de ocho bits. El bus de direcciones es de 16 bits, por lo que el microprocesador podrá direccionar 2 Exp 16=65536 direcciones distintas de memoria o sea 64 Kbytes (1 K=1024 bytes). Internamente el Z-80 posee celdas de memoria llamadas registros de CPU (unidad central de proceso). Estos registros se pueden conectar con la Unidad Aritmética y Lógica (ALU) que realiza operaciones como ser suma, resta y operaciones lógicas. También existe un decodificador de instrucciones y controlador de la CPU el que a través de microinstrucciones programadas le irá diciendo a la CPU cómo debe ejecutar el programa en código de máquina e irá ordenando los distintos momentos en que se deben realizar las operaciones. Además el microprocesador puede recibir del exterior señales de control a través del llamado bus de control, el cual nos referiremos más adelante.

Registros del Z-80

Como se puede observar en la Figura 2, el Z-80 posee 18 registros de 8 bits y 4 registros de 16 bits. El principal registro utilizado para el manejo de datos es el registro A. Este nombre proviene del término registro de acumulador, y que en él se "acumulan" los resultados de operaciones aritméticas. Si, por ejemplo, dos operandos deben ser sumados, uno de ellos se coloca en el registro A, luego, los operandos son sumados por la ALU y el resultado de la suma se vuelve a almacenar en el acumulador. Ya que este registro es de ocho bits, sólo se podrán guardar números decimales de 0 a 255 (2 Exp 8=256).

Además del registro A, el Z-80 posee otros registros de uso general designados como B, C, D, E, H y L; estos registros también son de ocho bits y se utilizan para el almacenamiento temporario de valores. Por ejemplo, un número que se encuentra en el acumulador puede ser almacenado en el registro E a través de la instrucción LD E,A (cargar E con A). De esta forma se tiene una ventaja con respecto a la

Figura 1 Diagrama de bloques del Microprocesador Z-80

Ambas computodaras poseen el misma microprocesador Z-80 de Zilog, al que le dedicaremos esto nata.

utilización ds msmoria axtsme: la CPU opera mucho más rápidamente con sus propios registros intsrnos

Como se pusde observer en le figure, los registros se pusden sgrupar de a pares: B con C, D con E y H con L, de ssta forme elgunes instrucciones del Z-8D permiten tratar e astos registros como ds 16 bits. Por sjemplo, se podrán realizar opereciones eritméticas con números ds 16 bits ds largo.

El registro F, que se puede epereer con sl acumulador A, ss llamedo Fleg (bandara). En esls registro queda elmacensda información ecerca ds los resultados de ciertas operacionss. De los ocho bits del registro se utilizan ssis Cuetro ds silos se pusdsn interrogar y utillzerlos como condición pera la ejecución de cisrtas condiciones. Eslos son le bandsra de acerreo (C), le bandera de csro (Z), la bandsra de signo (S) y le bandera de paridad/rebaesmiento (P/V). La ubicación de eslos bits dsntro del registro F puede observerss en le Figura 3.

Además de estos ocho registros generales existe un juego duplicado ds regisiros que se idsnlifican con ('). De todas formas sólo un jusgo eslá activo al mismo tiempo, ya sea AF, BC, DE, HL o A'F', B'C', D'E', H'L' pero el programedor tiens la posibilidad de intercambiarlos con las instrucciones de intercambio EX y EXX.

Por último sl Z-80 posee seis registros de propósitos especiales.

Contador de programe (PC): Este regisiro contisne la dirección (16 bits) ds le insirucción qus sn ese momsnto se sstá buscando en la msmoria. Este contedor es automálicemente incrementedo despuás de que cada byte es lsído. después de cada byte es leído.

Si se llsgara a encontrar un selto o llamado s subrutina antonces la nusva dirección se sitúa sn el PC.

Puntero de pila (SP): Es un registro de 16 bits que apunte e un sector de le memoria RAM que se encusntra reservade para sl STACK (pile). El stack está organizado de forma tal, que los datos puedsn ser elmecensdos y recuperados de le siguiente mansra: el último qus entra es el primero qus sele (LIFO last-in, first-out). Esle área ds memoria puede ser utilizada para guerder distintos valoras cusndo no se quieran usar los registros del microprocesador, pero además cumple una importante función, ya que en slla se elmecenen las direcciones de ratorno cuendo se ejecutan

Figure 2 Configuración de registros de la CPU Z-80

VECTOR INTERRUPCION	REGENERACION DE MEMORIA R	REGISTROS DE USO ESPECIAL
REGISTRO INDICE IX		
REGISTRO INDICE IY		
PUNTERO DE PILA SP		
CONTADOR PROGRAMA PC		

CONJUNTO REGISTROS PRINCIPALES

ACUMULADOR	INDICADORES DE ESTADO
A	F
B	C
D	E
H	L

REGISTROS DE USO GENERAL

CONJUNTO REGISTROS ALTERNATIVOS

ACUMULADOR	INDICADORES DE ESTADO
A'	F'
B'	C'
D'	E'
H'	L'

instrucciones de llamado a snbrutina o se reloma de una inlarrupción

Registros indice (IX, IY). Son registros de 16 bits qne permilan un lipo de direccionamianlo da memona llemado "Indexado". Su utilización es mny común an al manajo de tablas.

Vector de Inlarrupcionea (I) y Registro de Refresco (R): El ragialro I se ulilza cusndo el modo de inlarrupciones vectonzadas es elegido: suminisira el byta ello de la dirección en la cual se realizará un llamado indirecto a una rulina de atención de Inlerrupciones. El registro R se ulilza pera el refresco de mamonas RAM dinámicres.

Señelea de Entrade-Selide

Pesamos ahora a considerar como el Z-80 se comunica con al mundo extamo estndisndo las conaxiones elécincas del mismo.

La Iigura 4 muasira la función da los distinlos pines, sal como lambién su designación en el "chip". La lensión de allmanlación del Z-80 as da 5 volls y todas las entradas y salides son TTL compsl ibles, lo que signilica que un 0 lógico es aproximadsmenle 0 volls y un 1 lógico eslá enire 3 y 5 volls.

La antrada de reloj es una onds cuadrada da 3,528 MHz que se obliane da un osollador Inlerno en smbas computadoras. Cada Instrucción que ejecula al Z-80 se divide an los llamados ciclos "T"

En la Fignra 5 se puade observar como un ciclo da Inslrucción se encuentra dividido en varios ciclos T. En esla ajemplo se está ejecutando la Inslrucción INC (HL), en la cual la CPU lee el valor da HL, Incramanla en uno y lo vualve a almacenar an HL. En la primara parle dal ciclo de máquina la CPU realiza nna brisqueds del código de operación, an la que Inlerpreta la inslrucción an código de máquina, en los siguienles ciclos raalizará la lectura, incremento y nneva escritura del regislro HL

Pasemos ahora a describir brevemanla las tunclonas de loa demas pinas.

A0-A15 · Bus da direcclonas. Es tipo tri-esledo, sctivo allo. Provee las direcciones para el Inlarcambio de información con la mamoria e disposilivos de antrada-salida (I/O).

D0-D7 - Bus de dalos - Estas ocho lineas bidireccionales Iri-eslado son ntilizadas para la lransferencia de dalos con le CPU. La Iransferencia tiana lugar anlre la mamoria o los penféncos y el Z-80.

M1 · Ciclo de máquine 1 - Salids activa-baja. Esla sañal significa qne la CPU está obteniendo el código de operación para la próxima Inalrucción a ser ejecutada M1 lambién apareca con IORO para Indicar un ciclo de reconocimianlo de Interrnpciones

MREQ · Mamory requast · Salida tri-eslado, acliva-baja. Esta señal la Indica a la mamoria qua al bus de direcciones conliene una dirección válida para realizar una operación de leclura o ascritnra.

IORQ - Input / oulput requesl · Salida tri-estado, activa-baja. Indica que la mitad baje del bus da direcciones conllene una dirección válida de entrada-salida. Esla señal se ntilize para distingnir anlra una operación de anlrads-salida o una oparación qua Involucra a la memoria. IORO y MREO nunca eslarán ectivas al mismo llampo.

RD · Read · Selida tri-eslado, activa-baja. Indica qua la CPU realizsrá una leclnra da memorle o dasda los disposilivos de sntrada-salida

WR · Write · Idam anlerior, pero la CPU realizará una ascritura.

RFSH · Refresco · Salida, activa-baje. Indica qna los sialo bits menos significalivos del bus de direcciones coaliənan une dirección de relrasco para memorias dinámicas.

HALT · Salida, activa-baja. Indica

Figure 3 Posicionea de loa bita en el registro de estedo

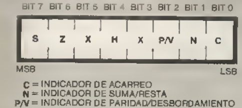
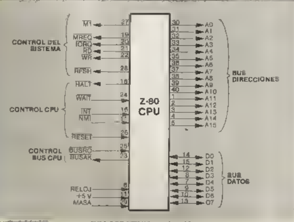


Figure 4 Terminalea (pioas) en el Micro Z-80



que la CPU ha ejecutado una detención por software provocada por una instrucción denominada HALT. La CPU esperará una interrupción antes de continuar su operación

WAIT · Entrada, activa-baja. Indica a la CPU que la memoria o los dispositivos de entrada-salida direccionados no están aún listos para una transferencia de datos. La CPU no realiza ninguna operación hasta que el dispositivo lento le indique que está llsto.

INT · Pedido de interrupción · Entrada, activa-baja. Esta señal se genera en dispositivos externos ordenándole a la CPU que corra un programa especial en código de máquina ubicado en un lugar determinado de la memona

NMI - Interrupción no enmascarable. Es una entrada operada por el flanco descendente de la señal. La misma fuerza a la CPU a correr un programa en código de máquina que comienza en la dirección decimal 102. A diferencia del otro tipo de interrupción, ésta no puede ser ignorada por la CPU. No se utiliza en estas dos computedoree

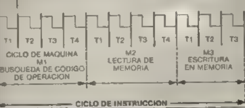
RESET · Entrada, activa-baja. esta señal inicializa la CPU borrando los registros PC, I y R. La CPU comenzará a ejecutar el programa a partir de la dirección 0 luego de un RESET

BUSREQ · Solicitud de bus. Entrada, activa-baja. Es utilizada por un dispositivo externo a fin de solicitar el control de los buses. La CPU pasará sus buses al estado "tri-estado" cuando se termine el ciclo de máquina que se estaba ejecutando.

BUSACK · Salida, activa-baja. Indica al dispositivo solicitante, que puede tomar ahora el control de los buses.

De esta forma hemos dado un rápido vistazo al Z-80. Para interiorizarse más sobre el funcionamiento de este microprocesador, existe bibliografía disponible en librerías técnicas. **L. MATARRESE G.E.**

Figura 5 Diagrama de tiempos durante una instrucción típica





RALLY 2025

COMP TS 1000/1500 TK 83/85
CONF 16 K
CLAS ENT
AUTOR Ricardo Alfredo Lucero
Córdoba





Al terminar de ingresar el programa en la computadora por primera vez, hay que hacerlo correr mediante RUN 4000 (RUN sólo no funcionará). La computadora se pondrá en modo FAST durante unos 120 segundos, regresando luego a SLOW para comenzar el juego.

La rutina 4000 se encarga de formar y guardar en memoria las pantallas que intervendrán en el juego. Por esto cuando vayamos a grabar el programa podemos borrarla para ocupar menos cinta, pero deberemos cuidarnos de no hacer CLEAR ni RUN. Grabémoslo mediante GOTO 4900.

En este juego nos disponemos a correr una etapa del "Rally del Futuro"

El piloto es el computador y nosotros los navegantes.

Por consiguiente deberemos darle las órdenes mediante teclado; y disponemos de los siguientes comandos

- ACELERAR (produce un aumento de velocidad de 10 ó 15 kilómetros): Tecla "A".
- DISMINUIR (disminuye la velocidad en igual proporción): Tecla "D"
- FRENAR (reduce la velocidad en 60 kilómetros): Tecla "F".
- MARCHAS: Teclas "1" a "5"
- GIRAR A LA IZQUIERDA: "Tecla "7".
- GIRAR A LA DERECHA: Tecla "0"
- DESVIARSE A LA IZQUIERDA: Tecla "8"
- DESVIARSE A LA DERECHA: Tecla "9"

Para dar arranque pulsamos una tecla

Deberemos controlar las revoluciones por minuto (RPM), las que no deberán bajar de 3500 (excepto en 1ª), ni superar las 6500. En el primer caso se apaga el motor con

```
   1 RETURN
  10 LET V=3
  20 LET TR=0
  30 LET K=0
  40 LET T=400
  50 LET A=0
  60 LET B=0
  70 LET N=1
  80 LET N1=1
  90 LET S=0
 100 LET B=1
 110 LET C=1
 120 PRINT AT 0,0,B$(1
 130 PRINT AT 0,16, [illegible]
[illegible]
 140 IF INKEY$="" THEN GOTO 140
 150 PRINT AT 2,16 [illegible]
[illegible] AT 2,16, [illegible]
 180 IF INKEY$ [illegible] THEN GOTO 150
 190 PRINT AT 2,10, [illegible]
[illegible]
 200 LET N=3
 210 GOTO 230
 220 LET N=INT (RND*6)+3
 230 GOSUB 100*N
 240 LET C=1
 250 LET TR=TR+INT (T/50)
 260 LET K=K+0
 270 IF S=1 THEN GOTO 1940
 280 GOTO 220+1640*(B>35 AND N<1
)+1660+(B/65)+1060*(K>9)
 300 GOSUB 1100+30*(C>1)
 310 GOSUB 1020+(A$="7" OR A$=
0 ) AND A(0)+1550*(A$= 9 )+1300*
(A$="A )+1350*(A$= D )+1400*(A$=
F )+1440*(CODE A$ =28 AND CODE
A$>=33)
 320 LET C=C+1
 330 IF S=1 OR C<0 THEN RETURN
 340 GOTO 300
 400 LET B=2
 410 GOSUB 1100+30*(C(1)
 420 GOSUB 1060*(A$ ("7" AND A$(
 0  AND A$(   AND C=3)+1620*(A
$= 0  OR (A$=   AND C=3 (+1550=)
A$= 0  AND C(3)=1400*(A$= 7')=15
00+(A$= A  AND C>3 =1550*(A$='D'
 AND C 3(+1400*(A$= F" AND C 3(=
1440*(CODE A$>=28 AND CODE A$<=5
3 AND C 5)
 430 LET C=C+1
 440 IF S=1 OR C>3 THEN RETURN
 450 GOTO 410
 500 LET B=3
 510 GOSUB 1100+50*(C(1(
 520 GOSUB 1000*(A$( 7" AND A$
( 0' AND A$<(   AND C=3(+1620=(0
$= 7" OR (A$=   AND C=3()+1550+(
A$='0' AND C>5)+1400+(A$= 0 (+12
00+(A$= A  AND C(3(+1300+(A$= D
 AND C<3)+1400+(A$= F  AND C D)=
1440+(CODE 0$(=28 AND CODE 0$(=3
5 AND C(3)
 550 LET C=C+1
 540 IF S=1 OR C(3 THEN RETURN
 550 GOTO 510
 600 LET B=4
 610 GOSUB 1100*50*(C(1)
 620 GOSUB 1550=(A$='9 )+1400*(A
$= 7  OR A$= 8 )+1300+(A$= A')+1
350*(A$= D (+1400*(A$='F )+1440*
(CODE A$(=28 AND CODE A$(=35(
 630 LET C=C+1
 640 IF S=1 OR C(3 THEN RETURN
 650 GOTO 610
 700 LET B=4
 710 GOSUB 1100+30*(C(1)
 720 GOSUB 1600+(A$>)'7' AND A$(
( 8' AND A$<('  AND C=5)+1620*(A
$=   AND C=3)+1550*(A$= 0  AND C
<3)=1400*(A$='7' OR A$='0")+1300
4)A$= A  AND C(3)+1350*(A$= D  A
ND C(3)+1400*(A$= F  AND C(5)+14
40*(CODE 0$)=29 AND CODE A$(=33
AND C(5)
 730 LET C=C+1
 740 IF S=1 OR C(3 THEN RETURN
 750 GOTO 710
 800 GOSUB 1100+50*(C>1)
 810 GOSUB 1620*(A$= 7" OR A$= 0
)=1650*(A$='0 )+1300*(A$= A )+1
550*(A$= D (+1400*(A$= F )+1440*
)CODE A$)=29 AND CODE A$ =53(
 820 LET C=C+1
 830 GOTO 840+40*(S=1)-40*(C<4 A
ND S(>1)
 840 LET N1=0+(A)80 AND A(130)+2
*(A(=130)
 850 LET N=INT (RND*N1)
 860 IF N=1 THEN GOSUB 1000
 870 LET I$= LA VELOCIDAD NO ERA
 LA APROPIADA- DA ARSA ARSAS UN VA
DO
 880 RETURN
 900 GOSUB 1100+50*(C(1)
 910 GOSUB 1600=(A$='7  OR A$= 8
)+1690*(A$= 8  AND A=1)+1520*(A
$= 0" AND A()1)+1550*(A$= 0 (+13
00+(A$= 0 )+1550*(A$= D )=1400+(
A$= F (+1440+(CODE A$ =29 AND CO
DE A$<=35)
 920 LET C=C+1
 930 GOTO 940+20+(S=1 OR A$="0
OR A$= 9 )-40*(C(4 AND S>(1)
 940 GOSUB 1000
 950 LET I$= LA FIEDAR ROMPIO EL
 EJE DEL AU- TOMOVIL
 960 RETUON
1000 GOSUB 1100+50*(C(1)
1010 GOSUB 1620*(A$="7  OR A$='0
 (+1520*(A$= 0 )+1550*(A$='9 (+1
300*(A$="A )+1350+(A$="D )+1400+
(A$= F )+1445+(CODE A$(=28 AND C
ODE A$>=33(
1020 LET C=C+1
1030 GOTO 1040+20+(S=1 OR A$= 0
 OR A$= 9 )-40*(C(4 AND S<>1)
1040 GOSUB 1000
1050 LET I$="NO INTENTE EMULAR A
 LOS HEROES  DE LA TV, OBSAETE L
OS CARTELES  DE LA RUTA
1060 RETURN
1100 PRINT AT 0,0,B$(0),AT 0 4,A
, [illegible] AT 0,9,M,AT 0 14,A, 00 ,AT
0 21 K,AT 0,26 TR,AT 0,31,U
1110 IF H(0 THEN GOSUB 1055+0+2
1120 LET B=1
1130 PRINT AT C 0, [illegible]
1140 PAUSE T
1150 LET A$=INKEY$
1160 PRINT AT C 0, [illegible]
1170 RETURN
1200 PRINT AT 6 15, [illegible] AT 7,14
 [illegible] AT 8,13,
1210 RETURN
1220 PRINT AT 8,0 [illegible]
[illegible]
1230 RETURN
1240 PRINT AT 10 19, [illegible] AT 10,
19, [illegible]
1245 LET A=INT (RND+2)
1250 IF A=1 THEN PRINT AT 17,9,
[illegible] AT 18 8 [illegible]
1255 RETURN
1260 PRINT AT 14,16    MANO
AT 15 16,  CONTRA   AT 16 17,
[illegible] AT 17 17, [illegible]
1290 RETURN
1300 LET A1=INT (RND*2)
1310 LET A=A+10*(A1=0)+15*(A1=1
1320 LET T=T-20*(A1=0)-50*(A1=1
1325 LET A=INT (A/M1)
1330 PRINT AT 0,4,A [illegible] AT 0,14,
A, 00
1340 RETURN
1350 LET A1=INT (RND+2)
1360 LET A=A-10*(A1=0 AND A(9)-1
5*(A1=1 AND A>14)
1370 LET T=T+20=(A1=0 AND A>0)+3
5*(A1=1 AND A(14)
1380 LET B=INT (A/M1)
1385 PRINT AT 0,4,A, [illegible] AT 0,14,
A 00
1390 RETURN
1400 LET A=A-60*(A(56)
1410 LET T=T+120*(A,50)
1420 LET A=INT (A/M1(
1425 PRINT AT 0,4 A, [illegible] ,AT 0,14,
A 00
1450 RETURN
1440 LET M=VAL A$
1450 LET N1=0/8+.5
1455 LET A=INT (A/01)
```

la consiguiente pérdida da tiempo. En el segundo se funde el motor, con lo que se pierde un auto.

Al presentarse la situación tendremos la posibilidad de ejecutar hasta tres maniobras, y seis en recta. Pero si realiza la maniobra principal de la situación (por ej. doblar si hay una curva) en la primera oportunidad pasará de inmediato a la situación siguiente.

La computadora indica que está esperando la orden mediante una señal arriba a la izquierda. Al encenderse dicha señal digitemos la orden correspondiente. Si la señal se apaga antes de que hayamos pulsado algo habremos perdido esa oportunidad y vendrá la siguiente (si es que no era la última). El tiempo de espera de la computadora varía de acuerdo a la velocidad.

Las situaciones que se pueden presentar son: recta, curva, cruce o bifurcación (en ambas es válida cualquier dirección), vado (representado por una banda gris que cruza la ruta), piedras (manchas grises sobre la misma), o mano cortada.

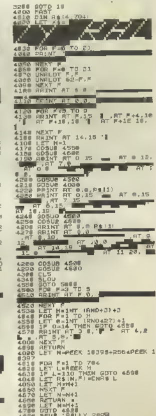
Si perdemos los tres vehículos no podremos completar la etapa. En caso contrario encontraremos entre los 150 y 250 kilómetros la llegada. En ambos casos podremos observar los resultados finales, donde se indicarán los kilómetros recorridos, el tiempo empleado, el promedio y el puesto clasificatorio (esto último sólo si completamos la etapa).

A continuación nos preguntarán si deseamos correr otra etapa. Si contestamos que sí (S) nos pedirán nuevamente el nombre (pudiendo pulsar ENTER); y si contestamos no (N) aparecerá la pantalla final y se detendrá en un estado en el que sólo responderá a BREAK.

NOTA. La justificación de la línea 1 es que a causa de los GOSUBs condicionados que contiene el programa, cuando se pulsa una tecla equivocada, o en algunos casos en los que no se pulsa nada, se ejecuta un GOSUB 0. **K64**

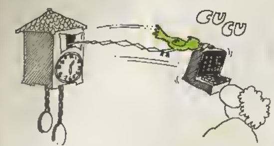
```
1460 PRINT AT 0,0,M,AT 0,14,R "0
0"
1470 RETURN
1480 LET N1=2*(R)110 AND R<140)+
2*(R)=140)
1490 LET N=INT (RND*N1)
1500 IF N=1 THEN GOSUB 1000
1510 LET I$="LA CURVA FUE TOMADA
 A EXCESIVA  VELOCIDAD
1520 LET C=3
1530 RETURN
1550 LET N1=2+(R)100 AND R<130)+
2*(R)=130)
1560 LET N=INT (RND*N1)
1570 IF N=1 THEN GOSUB 1000
1580 LET I$="RIDO LA BANQUINA A
MUY ALTA VE- LOCIDAD"
1590 LET C=C+2*(H=9 OR H=10)
1600 LET TR=TR+1
1610 RETURN
1620 GOSUB 1000
1050 LET I$= TOMO LA DIRECCION E
QUIVOCADA.   NO INTENTE SEGUIR P
OR DONDE NO  HAY CAMINO"
1040 RETURN
1600 GOSUB 1000
1070 LET I$="?????????????????"
1000 RETURN
1090 GOSUB 1000
1700 LET I$="LAS PIEDRAS ESTABAN
 TAMBIEN SO- BRE LA CONTRAMANO,
EN UNA PRO-  XIMA SITUACION SIMI
LAR OBSERVE  MEJOR ANTES DE TIRA
RSE A LA IZ- QUIERDA
1710 RETURN
1800 LET V=V-1
1805 LET S=1
1802 FOR G=0 TO 20
1210 FAST
1020 LET I=INT (RND*21)
1030 LET J=INT (RND*52)
1040 PRINT AT I,J,"
1045 SLOW
1050 NEXT G
1055 RETURN
1600 LET I$="LAS REVOLUCIONES BA
JARON Y SE   AHOGO EL MOTOR"
1050 LET TR=TR+2
1090 GOTO 1940
1900 LET I$= UD NO CONTROLO DEBI
DAMENTE LAS  REVOLUCIONES Y FUND
IO EL MOTOR"
1920 LET V=V-1
1950 LET TR=TR+2*(V)0)
1940 CLS
1945 PRINT AT 7 0,"INFORME DE LA
 COMPUTADORA","-----------------
---------"
1050 LET X=10
1900 LET Y=0
1970 FOR G=1 TO LEN I$
1900 PRINT AT X,Y," ",AT X,Y,I$(
G),
1090 LET Y=Y+1
2000 IF Y=32 THEN GOTO 2060
2010 NEXT G
2020 PRINT AT X,Y,
2030 FOR G=0 TO 20
2040 NEXT G
2050 GOTO 40*2160+(V=0)
2060 LET X=X+1
2070 LET Y=0
2050 GOTO 2010
2100 PRINT AT 0,0 P$(1),AT 1 0
                  AT 2 0
,AT 4,0      ,AT 5 0
```

```
2110 FOR F=5 TO 10
2115 PRINT AT F,5,   AT F,21,
2120 NEXT F
2130 PRINT AT 18 0
2140 FOR F=0 TO 50
2145 NEXT F
2150 LET C$="CANALETA
2160 GOTO 2300
2200 PRINT AT 0 0,P$
2270 PRINT AT 1,0
2240 FOR F=0 TO 50
2250 NEXT F
2260 LET C$="INCOMPLETA
2500 CLS
2310 PRINT "RESULTADOS FINALES
"------------------",,"PILOTO
   Z-80" ,"NAVEGANTE      ,N$, C
ONDICION  ETAPA  ,C$  KILOMETROS
 RECORRIDOS   ,K,"TIEMPO EMPLEAD
O   " TR
2320 LET A=INT ((K/TR)*50)
2330 LET AC=0
2540 FOR F=1 TO 10
2350 IF AC)0 THEN GOTO 2370
2300 LET AC=F*(A =145-10*F AND A
)=135-10*F)
2370 NEXT F
2300 LET O$=STR$ AC
2390 IF O$= 0   OR CODE C$=46 THE
N LET O$= NO CLASIFICO"
2400 PRINT "PROMEDIO   ,A,  KM/H
", PUESTO CLASIFICATORIO   ,O$
2410 PRINT AT 20,0. DESEA CORRER
 OTRA ETAPA PARA    MEJORAR SU T
IEMPO?(S/N)
2420 GOTO 2420+80*(INKEY$="N")+2
410*(INKEY$="S")
2500 CLS
2510 PRINT AT 0 0 X$(1 TO 32)
$(1 TO 64), X$(1 TO 96)  X$(1 TO
 128), X$(33 TO 128)  X$(65 TO 1
28), X$(97 TO 128)
2520 GOTO 2520
3000 PRINT AT 0 0 P$(1
3010 FOR F=0 TO 5
3020 PRINT AT F 0
3030 NEXT F
3040 PRINT TAB 7,
3050 FOR F=0 TO 2
3060 PRINT TAB 7
 TAB 7,
3070 NEXT F
3080 PRINT TAB 12,
3090 FOR F=0 TO 20
3100 PRINT AT F,1,
3110 NEXT F
3120 PRINT AT 1 0,X$,AT 10,11
                    AT 17,11,
                    AT 18,11,
                    AT 19,11,
3130 FOR F=0 TO 80
3140 NEXT F
3150 CLS
3160 PRINT AT 21 0,
3170 INPUT M$
3180 IF N$()    THEN LET N$=M$
3190 LET D=INT (RND*100+150)
```



```
3200 GOTO 10
4000 FAST
4010 DIM A$(4,704)
4020 LET A$=
4030 FOR F=0 TO 21
4040 PRINT "
4050 NEXT F
4060 FOR F=0 TO 31
4070 UNPLOT F,F
4080 UNPLOT 62-F,F
4090 NEXT F
4100 PRINT AT 5 0
4110 PRINT AT 0 0,
4120 FOR F=0 TO 9
4130 PRINT AT F,15    ,AT F+4,10
      AT F+10,10      AT F+1E 10,
1
4140 NEXT F
4160 PRINT AT 14,15 "
4100 LET H=1
4170 GOSUB 4550
4180 GOSUB 4600
4190 PRINT AT 0 15        AT 0 12,
      ,AT 7,0
  AT 0 0                      AT
0 0,
4200 GOSUB 4500
4210 GOSUB 4000
4220 PRINT AT 0.0,P$(1)
4230 PRINT AT 0,15        AT 0,15
      ,AT 7 15
     AT 6,15,
AT 10,19
4240 GOSUB 4500
4250 GOSUB 4600
4200 PRINT AT 0,0 P$(1)
4270 PRINT AT 0,0
 ,AT 0,10,                   ,AT 0
12,             AT 10,0
      AT 14,19                 AT
 1_ 0                AT 11 20,
4200 GOSUB 4500
4290 GOSUB 4600
4300 CLS
4340 SLOW
4550 GOTO 5000
4500 FOR F=3 TO 5
4510 PRINT AT F,0,
4520 NEXT F
4530 LET H=INT (RND+3)+3
4540 FOR F=1 TO H
4550 LET O=INT (RND*27)+1
4560 IF O=14 THEN GOTO 4550
4570 PRINT AT 3 0,"     AT 4,0
    ,AT 5,0,"
4000 NEXT F
4090 RETURN
4000 LET N=PEEK 16398+256*PEEK 1
6397
4610 FOR F=1 TO 704
4620 LET L=PEEK N
4630 IF L=110 THEN GOTO 4690
4040 LET A$(N,F)=CHR$ L
4050 LET N=N+1
4660 NEXT F
4670 LET N=N+1
4680 RETURN
4590 LET N=N+1
4700 GOTO 4620
4800 SAVE "RALLY 2020"
5000 GOTO 3000
```

RELOJ DIGITAL



Si desafortunadamente se nos ha roto el reloj, con este programa bien cortito podremos disponer de un reloj digital. Tal vez no resulte muy práctico, pero funciona. Deberemos arrancar con RUN.

COMP. CZ 1000/1500° TK 83/85
CONF 2 K
CLAS. UTI

```
  1 REM RELOJ DIGITAL
  4 LET L=1
  5 PRINT "HORA ?"
 10 INPUT H
 15 PRINT "MINUTO ?"
 20 INPUT M
 25 CLS
 35 IF M=60 THEN GOTO 100
 39 PRINT AT 10 10 H;" " M
 50 LET M=M+L
 60 FOR F=L TO 2732
 70 NEXT F
 80 GOTO 30
100 LET M=0
105 LET H=H+L
110 IF H>12 THEN LET H=
120 CLS
130 GOTO 39
```

SUBMARINO II



Estamos conduciendo un submarino y deberemos atravesar el mar esquivando las peligrosísimas minas. Se arranca con RUN y se maneja con las flechas.

COMP CZ 1000/1500, TK 83/85
CONF. 2 K
CLAS. ENT

```
  I REM SUBMARINO II
  ▮ FOR F=SGN PI TO INT PI
 10 CLS
 15 FAST
 2▮ PRINT --.---------------.---.-
............... AT VAL "8 NOT PI,
 2▮ FOR M=NOT PI TO CODE  2
 30 PRINT AT VAL "6 +RND+SGN PI
,RND+CODE "? +CODE ▌  +
 35 NEXT M
 40 SLOW
 4▮ LET D=INT (VAL  6 +RND+SGN
PI)
 ▮▮ FOR S=NOT PI TO CODE ",
 65 PRINT AT D S,
 6▮ IF PEEK (PEEK 16398+PEEK 16
399*256)=CODE "+ THEN PRINT "
TED NOT PI,"DESTRUIDO ",F AT D
3,
 65 PRINT ●
 7▮ LET A=D
 75 LET D=D+(INKEY$= 6" AND D<V
AL "7 )-(INKEY$= 7  AND D>SGN PI
)
 6▮ PRINT AT A,S,
 ▮5 NEXT S
 ▮▮ NEXT F
 9▮ PRINT "BIEN, ATRAVESASTE EL
PELIGRO
```

RENUMERADOR

COMP.' CZ 1000/1500, TK 83/85
CONF 2 K
CLAS. UTI



Con este programa de pocos pasos, se puede poner un poco de orden en nuestros revueltos programas. Sobre todo en aquellos que por teclearlos en pasos de a uno, no nos queda lugar en el medio de dos pasos para poder meterse. Se arranca con GOTO 9000.

```
▮▮▮0 REM RENUMERADOR
▮▮▮1 LET IO=1▮
▮▮1▮ FAST
▮▮2▮ LET L=16511
▮030 IF PEEK (L-2)|=35 THEN GOTO
▮1▮▮
▮▮40 LET M=PEEK L+256*PEEK (L+1)
▮▮5▮ POKE (L-2),INT (IO/256)
9▮6▮ POKE (L-1),IO-INT (IO/256)*
256
9▮70 LET IO=IO+1▮
9▮▮0 LET L=L+M+4
9▮90 GOTO 9030
9100 SLOW
9110 LIST
```

CONSTRUCTOR

COMP. CZ 1000/1500, TK 83/85
CONF. 2 K
CLAS. ENT



Deberemos empujar los signos "=", apretando cualquier tecla, y apilarlos línea a línea. Si soltamos la tecla, el signo se detiene, y ubicando el cuadrado negro, sabremos donde nos encontramos. Para cargarlo tardaremos menos de cinco minutos, pero horas para jugarlo. ¡Animo!

```
  1 GOTO I▮
  2 SAVE "CONSTRUCTOR"
 I▮ FOR I=▮ TO 21
 20 PRINT AT I,▮5 ■"
 3▮ FOR J=▮ TO ▮3
 4▮ PRINT AT I,J,"|%" AND INKEY
$<>""
 50 NEXT J
 6▮ PRINT AT I,▮▮
 70 NEXT I
 ▮▮ GOTO ▮
```

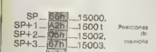
DESENSAMBLANDO EL ASSEMBLER

Eduardo H. Mombello PARTE II

Veremos ahora el último de los registros y sus dos instrucciones asociadas, antes de comenzar a programar.

El registro SP (Stack Pointer)

El SP o puntero de la pila, es un registro un tanto especial de 16 bits, veamos su funcionamiento.
La pila a la que hace referencia es un conjunto de bytes consecutivos, que los podemos pensar como cajitas una encima de otra, de ahí el concepto de pila de bytes.
Entonces la cajita que se encuentra en la cima de la pila será la posición de memoria de menor valor, y el SP contendrá el valor de dicha posición, aclaremos esto,



En el cuadro anterior vemos que la "pila" comienza en la posición de memoria 15000, y que el SP apunta a esta dirección, también se ve que dicha posición contiene el valor 56 hexadecimal.
Cabe preguntarnos para qué sirve ésto, pues bien, se lo utiliza para guardar datos que nos servirán más adelante justamente apilados como si fueran libros.

PUSH y POP

Por fin llegamos a ver nuestras primeras instrucciones, hablemos de PUSH.
Esta nos permite guardar datos contenidos en cualquiera de los pares BC, DE, HL, AF, IX e IY, en la cima de la pila a la que apunta el SP.
Un dato importante es que el contenido del registro que guardemos en la "pila" permanece en dicho registro sin ser afectado.
Veamos un ejemplo:
Supongamos que nuestro SP está en las condiciones del cuadro que vimos antes, y que el par DE contiene el número 16384, o sea por lo que vimos en el número anterior el registro D contendrá el valor 64, y el E tendrá un 0.
Entonces veamos cómo quedará nuestra "pila" una vez que ejecutemos la instrucción PUSH DE.

```
SP ——  [ 0. ]  14998.
SP+1–  [ 64 ]  14999     Posición
SP+2–  [56h ]  15000     de
SP+3–  [A2h ]  15001.    memoria
,P+4–  [05h ]  15002.

      "pila"
   [D | 64.]   Contenido
   [E |  0.]   del
               par DE.
```

Ahora tenemos una copia del contenido del par DE encima de la pila, notar que nuestra pila comienza ahora en la posición de memoria 14998, con lo cual si examinamos el contenido del SP veremos que justamente ha tomado ese valor.
Otra cosa importante de observar es que guarda "primero" el contenido del registro de menor peso y luego el de mayor.
En nuestro caso, el contenido de E se encuentra en el lugar a donde apunta SP, y el de D está a donde apunta SP+1.
A partir de este momento cuando encerremos entre paréntesis a cualquiera de los registros dobles o a un número que represente una posición de memoria, estaremos refiriéndonos al comienzo de dicha posición de memoria o al contenido de la posición de memoria que contiene el registro.
Ejemplo: si el par BC contiene el número 65535, y en la posición de memoria 65535, se encuentra el número 12, se tendrá.
BC=65535 (BC)=12, y
(BC)=(65535.)=12
Si, en cambio, ahora, queremos guardar en uno de los registros dobles lo que se encuentre en la cima de nuestra pila, que en definitiva es extraer el contenido de los dos bytes superiores de la pila, debemos ejecutar la instrucción POP rr, donde rr es cualquiera de BC, DE, HL, AF, IX e IY.
Como ejemplo podemos ver que, si tenemos la memoria de nuestro computador cargada con los valores del ejemplo anterior, al igual que el SP apuntando a la posición 14998., luego de hacer la extracción por medio de POP HL el SP apuntará a la posición 15000.
El par HL tendrá el valor 16384., entonces nuestra representación de las celdas de memoria será:

```
No
pertenecen
el
stack.   [ 0. ] 14998.
         [ 64 ] 14999     Posición
SP       [56h ] 15000     de
SP+1     [A2h ] 15001.    memoria
SP+2     [05h ] 15002.
```

Figura 1

Posición de memoria	CM	Mnemónico	Notas
40001	245	PUSH AF	Guarda los registros
40002	197	PUSH BC	en la pila
40003	213	PUSH DE	
40004	229	PUSH HL	
40005/6	62 170.	LD A,170.	A 170.
40007/8/9	33. 110. 75	LD HL,19310.	HL 19310.
40010	119	LD (HL),A	19310. A
40011	225	POP HL	Extrae los valores
40012	209	POP DE	de los registros
40013	193	POP BC	de la pila
40014	241	POP AF	
40015	201.	RET	Retorna al BASIC

En esta nota terminamos de explicar los registros y empezamos a ver algunas instrucciones.

Contenido del par HL.	H 64
	L 0

Entonces podemos decir que POP hace el camino inverso al de PUSH.

Grupo de carga de 8 bits

Antes que nada dejaremos en claro algunas cosas:

I) Ninguna de las instrucciones que pertenecen a este grupo afecta al indicador de estado F (flag).

II) r y r' son cualquiera de los registros A, B, C, D, E, H o L.

III) n es un número cualquiera representable en 8 bits (0 a 255.).

IV) nn es un número representable en 16 bits (0. a 65535.).

V) (nn) indica el contenido de la posición de memoria representada por nn.

VI) El CM (código máquina) de cada instrucción se dará a medida que lo requieran los ejemplos, de todos modos en el manual de la TS 2068 pueden encontrarlos en el apéndice B.

Aclarados estos puntos comencemos con las instrucciones.

LD r,r': ésta copia el contenido de r' en el registro r.

LD r,n: r toma el valor n.

LD r,(HL): r toma el valor que está guardado en la posición a la que apunta HL.

LD (HL),r: copia en la posición de memoria a la que apunta HL el número que contiene r.

LD (HL),n: copia n en (HL).

LD A,(BC): copia en el acumulador (BC).

LD A,(DE): copia en el acumulador (DE).

LD A,(nn): copia en el acumulador (nn).

LD (BC),A: copia en (BC) el contenido de A.

LD (DE),A: copia en (DE) el contenido de A.

LD (nn),A. copia en (nn) el valor de A.

Estas son las que más nos interesan en este grupo, pues son las de uso más corriente.

Ahora sí, programemos algo y anticemos en detalle su funcionamiento.

En primer lugar debemos elegir un sitio seguro para alojar nuestro programa.

En el caso del Spectrum y de la 2068 existe una instrucción Basic (CLEAR) que seguida de un número correspondiente a una posición de memoria, le indica al sistema operativo que cualquier programa basic que se intente ingresar no podrá sobrepasar dicha posición de memoria.

Por ejemplo, si ingresamos CLEAR 40000 nos quedarán 25 kbytes después de la posición 40000, para guardar "cosas" que no sean un programa basic.

En nuestro caso lo usaremos para guardar el CM.

Es importante también, que leamos del manual de nuestra computadora cómo se encuentra repartido el lugar de trabajo del sistema operativo, por el momento busquemos el comienzo y fin de la memoria de pantalla.

En el caso del TS 2068 y del Spectrum ésta comienza en la posición 16384. y termina en la 22527.

En primer lugar, podemos ver que el programa de la figura 1 utiliza instrucciones que conocemos, veamos qué es lo que hacen.

Los cuatro primeros PUSH guardan los valores que tenían los registros antes de la ejecución de nuestra rutina en CM.

Luego, se carga el acumulador con el valor 170.(10101010) y se copia este número en la posición de memoria 19310.

Esta última, como podemos ver, se encuentra dentro del rango de memoria destinado a la pantalla.

Por último, los cuatro POP devuelven su antiguo valor a los registros, y RET finaliza nuestro programa entregando el control al sistema operativo del Spectrum ó 2068

Al ejecutar esta rutina (con RANDOMIZE USR 4e4) sólo veremos cuatro puntitos en medio de la pantalla (posición 19310.) provenientes del número 170.(10101010).

Lo importante es que nos detengamos a entender cada punto del programa (figura 1), y ver su forma de llevarlo a la práctica (figura 2) por medio del basic

Ahora veremos otra rutina (figura 3) que llenará la pantalla con "basura", pero que tiene por finalidad hacer la comparación del tiempo que tarda en Assembler un bucle (FOR) de 6911 pasos, con lo que tardaría en terminar el mismo bucle pero en basic.

Para ver este efecto compilemos el CM de la figura 3, de forma similar al de la figura 2, y ejecutémoslo (RANDOMIZE USR nn). Luego carguemos el basic de la figura 4 y ejecutémoslo (RUN).

Si prestamos atención a la ejecución del programa en basic, notaremos la forma en que se distribuyen los bytes de pantalla, es lento ¿no? **K64**

Figura 2

```
10 CLEAR 40000
20 FOR f=40001 TO 40015: READ a
30 POKE f,a: NEXT f
40 DATA 245,197,213,229,62,170,33,110,75,119,225,209,193,241,
   201
```

Figura 3

Posición	CM	Mnemónico
40001/2/3	33. 0. 0.	LD HL,0.
40004/5/6	17. 0. 64.	LD DE,16384
40007/8/9	1. 255. 26.	LD BC,6911
40010/11	237 176.	LDIR
40012	201	RET

Figura 4

```
10 FOR f=16384 TO 16384+6911: POKE f,PEEK (f-16384): NEXT f
```

EL SISTEMA TRIDIMENSIONAL

En el libro "Ideas y formas. Explorando el espacio con Logo", del ingeniero Horacio Reggini (Ed Galápago, Dist. Emecé) el autor explica la creación y el manejo de formas tridimensionales.

El sistema tridimensional básico es el más rápido y simple. Contiene los procedimientos necesarios para mover la tortuga tridimensional y realizar figuras espaciales (ANDAR, ROLAR, CABECEAR, ROLAR, TRI)

Esta versión del sistema no hace aparecer en la pantalla la tríade —conjunto de tres ejes que muestran la ubicación y orientación de la tortuga tridimensional—; tampoco incorpora las órdenes de movimientos relativos a los ejes coordenados fijos.

El sistema tridimensional completo, además de los procedimientos necesarios para mover la tortuga tridimensional y realizar figuras espaciales, tiene la capacidad de hacer aparecer en la pantalla la tríade. Esta versión del sistema incorpora a las órdenes del sistema básico aquellos procedimientos que permiten la manipulación de la tríade (TRIINV, TRIADA, SINTRIADA, VISIBLE, INVISIBLE). Las órdenes de movimiento, tanto del sistema básico como del sistema completo, están referidas a la geometría intrínseca de la tortuga tridimensional; no han sido incorporadas las órdenes de movimientos relativos a los ejes coordenados.

El sistema adicional incorpora órdenes espaciales referidas a la geometría extrínseca de la tortuga tridimensional; o sea, los procedimientos que definen las ubicaciones de la tortuga tridimensional en relación a los ejes coordenados (PX3D, PY3D, PZ3D, PXYZ3D, POS3D, CENTRO3D, PUNTO3D), que son similares a las usadas por la tortuga bidimensional en el plano.

Esta versión incluye también órdenes de rotación de la tortuga alrededor de los ejes fijos (GIRARX, GIRARY, GIRARZ).

SPECTRUM CZ-2000

```
TO ANDAR D
MAKE 'X (X  D* C11)
MAKE 'Y(Y+  * C1 )
MAKE "Z(Z+D* C13
GRAFICAR X Y Z
END

TO V      A
MAKE "C COS A
MAKE   IN A
MAK    "11  11* C + C21*
S)
MAKE   12     2* C+ C
S)
MAKE   13( C13*C+ C23
S)
MAKE '121) C21* C  C11*
S)
MAKE "T22( C22* C-- C12*
S)
MAKE "T23( C23*C-- C13*
S)
REEMPLAZAR
END

TO CABECEAR A
MAKE "C COS A
MAKE "S SIN A
MAKE "T11( C11* C--C31*
S)
MAKE "T12( C12*C--C32*
S
MAKE "T13( C13* C-- C33*
S)
MAKE "T31( C31*C+ C11*
S)
MAKE "T32( C32* C+ C12*
S
MAKE "T33( C33*C+ C13*
S
REEMPLAZAR
END

TO ROLAR A
MAKE "C COS A
MAKE "S SIN A
MAKE "T21( C21* C+ C31*
S)
MAKE "T22( C22* C+ C32*
S)
MAKE "T23( C23* C+ C33*
S)
MAKE "T31( C31* C-- C21*
S)
MAKE     2    2* C  C22*
S)
MAKE "T33( C33* C  C23*
S)
REEMPLAZAR
END

TO REEMPLAZAR
MAKE  C11 T11
MAKE "C12 T12
MAKE "C13 T13
MAKE  C21 T21
MAKE "C22 T22
MAKE   23 T23
MAKE "C31 T31
MAKE  C32 T32
MAKE "C33 T 3
END

TO GRAFICAR U V W
SETPOS SE ( F* U/( F-- W
)) ( F* V  F-- W))
END

TO TRI
SETBG 6 CS CT HT WINDOW
MAKE "F 360
MAKE "T11 0
MAKE "T12 1
MAKE "T13 0
MAKE "T21 --1
MAKE "T22 0
MAKE "T23 0
MAKE "  31 0
MAKE "T32 0
MAKE "T33 1
MAKE  X 0
MAKE "Y 0
MAKE     0
REEMPLAZAR
END
```

SPECTRUM CZ-2000

```
TO RECTANGULO
REPEAT 2 (ANDAR 10 VIRAR 90 AND
R 120 VIRAR 90)
END

TO MOLINETE
REPEAT 18 (RECTANGULO ROLAR 20)
END
```

Los procedimientos desarrollados por el ingeniero Horacio Reggini pueden ser implementados en lo mayoria de las microcomputadoras que operan con decimales.

Implementación del sistema en diferentes equipos



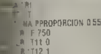
T-PC acepta el sistema completo procedimtentos relacionados tanto con la geometria intrínseca de la tortuga tridimensional como con la geometría extrínseca). las Caracteristicas particulares del sistema Logo implementado en esta máquina —fundamentalmente la orden PLUMADEMEZCLAR— permiten la rapresentación de la tríada en colores.

IBM-PC acepta el sistema completo (procedimientos relacionados tanto con la geometría intrínseca de la tortuga tridimensional como con la geometría extrínseca). La tríada puede ser representada con el mismo color usado para el dibujo. Línea Apple: puede operarse al sistema completo, con mayor o menor comodidad según la capacidad de memoria del equipo. La tríada se representa en el mismo color del dibujo.

COMMODORE 64 es simple la implementación del sistema básico (procedimientos relacionados con la geometría intrínseca de la tortuga tridimensional). Como la versión de Logo implementada en esta máquina no acepta la orden VENTANA, la figura debe circunscribirse a las dimensiones de la pantalla. Como tampoco exista la orden PLUMAINVERSA, no es factible realizar fácilmente la representación de la tríada.

SPECTRUM CZ-2000: se puede implementar fácilmente el sistema básico (procedimientos relacionados con la geometría intrínseca de la tortuga tridimensional). No existan restricciones al movimiento de la tortuga, puesto que acepta la orden WINDOW en el procedimiento TRI. Acoplando a la SPECTRUM la impresora térmica pueden obtenerse copias de los dibujos realizados **K64**

TRI·TI-PC 1/2



```
A 'RI
T
NA PPROPORCION 0 55
R F 750
R T1t 0
R 'T12 1
R T13 0
R T21 -t
R 22 0
ER T23 0
R T3t 0
F T32 0
R T33 t
MPLAZAR
R "X 0 HACER "Y 0 HACER 'Z 0

A GRAFICAR U V W
A ( F* U/( F-W)) ( F* V/( F-W))

A REEMPLAZAR
R Ctt Ttt
9 'C12 Tt2
R C13 T13
ER C21 T2t
R C22 T22
ER C23 .T23
R 'C3t .T3t
FR C32 T32
ER C33 T33

A DESOCUPAR
RAR ANDAR VIRAR CABECEAR
AR TRI GRAEICAR REEMPLAZAR
RAR NOMBRES
RAR DESOCUPAR
```

TRI-TI-PC 2/2

```
PARA ANDAR D
HACER "X( x+ D* Ct1)
HACER "Y( Y+ D* C12)
HACER "Z( Z+ D* Ct3)
GRAFICAR X Y Z
FIN

PARA VIRAR 'A
(LOCAL 'C "S)
HACER 'C COS A
HACER 'S SEN A
HACER 'T1t( C11* C+C21*·S)
HACER 'T12( Ct2* C+C22* S)
HACER 'Tt3( C13* C+C23* S)
HACER 'T2t( C21* C-C11* S)
HACER 'T22( C22* C-Ct2* S)
HACER 'T23('C23*·C-:Ct3* S)
REEMPLAZAR
FIN
PARA CABECEAR A
(LOCAL "C "S)
HACER "C COS A
HACER "S SEN A
HACER 'Tt1( C1t* C-:C31* S)
HACER "T12(,Ct2*,C-'C32*,S)
HACER "Tt3( C13* C-C33* S)
HACER 'T31('C31* C+:Ct1* S)
HACER 'T32( C32*·C+,C12* S)
HACER "T33('C33* C+Ct3*·S)
REEMPLAZAR
FIN

PARA ROLAR A
(LOCAL "C "S)
HACER "C COS A
HACER 'S SEN A
HACER "T2t( C21* C+'C31* S)
HACER "T22(:C22* C+C32* S)
HACER 'T23(.C23*.C+'C33*.S)
HACER "T3t('C31*.C-:C21*:S)
HACER "T32('C32*:C-'C22* S)
HACER "T33(.C33* C-:C23* S)
REEMPLAZAR .
FIN
```

Por primera vez en Argentina

K64 transmite en LASER 102

Programas para Home Computer
Todos los jueves a las 6,40 Hs.

TRANSMISION ENERO 1986

2/1/1986 CHECQUERED FLAG
La emoción de la formula 1 en tu computadora. Se te ofrece la posibilidad de correr en algunos de los mas famosos circuitos del mundo. El conducir un auto de estas características es una tarea de habilidad en la que se requiere evitar riesgos, permanecer en pista, cuidar el vehículo y lograr nuevos records.
Comandos
Acelerador Q
Freno 1
Rebaja N
Girar izquierda rápido A
Girar izquierda lento S
Girar derecha rápido F
Girar derecha lento D
Pausa H
Finalizar H y T a la vez
Hay que cuidarse de los siguientes riesgos: Agua y aceite en pista. Vientos. Recalentamiento del motor. Nivel de nafta, etc. Para los más inexpertos, se puede comenzar con el Mc Foster Special, que tiene caja automatica.

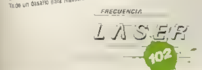
9/1/1986 MONOPOLIO
Este juego es parecido al tradicional "Estanciero". Permite el juego de 2 a 6 jugadores. Al comienzo cada uno recibe un informe del estado de caja y propiedades que posee.
Comandos
LIST Lista todas las propiedades que posee
H Le permite vender sus propiedades al banco o hipotecarlas
B Compra propiedades de otro jugador al precio convenido
S Vende propiedades a otro jugador
V Para recuperar propiedades del banco hipotecadas
Q Para el juego mostrando el jugador. Se puede en ese momento guardar el juego en cassette hasta donde se llegó
T Graba el programa en cassette
R En caso de haberse interrumpido el programa, arrancarlo con GOTO 0 y luego pulsar R
M Cuando un jugador hipoteca, recibe la mitad del valor de su propiedad. Para recobrarla con este comando se deducirá de sus reservas de caja esa cantidad mas un 10%
DOBLES Si sale dobles en los dados puede jugar otra vez pero si sale doble una tercera vez irá a la carcel permaneciendo en 3 jugadas. (Se puede salir pagando 5000)
Demás instrucciones durante el juego

16/1/1986 ANT ATTACK
La ciudad de "Antchester" ha sido abandonada por muchos años y permanece copada por gigantescas hormigas asesinas. Pero como siempre ocurre en toda aventura en esta no podían faltar los héroes que por primera vez en los juegos electrónicos en esta se puede elegir "heroe" o "heroína".
El objeto del juego es que "el" o "ella" rescate al otro buscándolo entre las ruinas de la ciudad y salir cuanto antes salvándose de las mordidas de las hormigas asesinas
Comandos
O P ENTER SPACE Cambia ángulo de visión
SYMBOL SHIFT Rota a reloj
M Rota contra reloj
V Avanza
C Salta
S D F G Tira granadas de distintos alcances
1 Ultima oportunidad, vuelve a la entrada

23/1/1986 OOPEDOORF
Este es un programa del tipo educativo, que permite realizar gráficos en baja, media y alta resolución de pantalla.
Es muy fácil de manejar y se muestran varios ejemplos logrados con él
Comandos
T Color de tinta
Q Ancho del trazo de 1 a 3
B Borra
C Acceso a líneas curvas
R Acceso a líneas rectas
O Borra pantalla
L Inserta texto
G Cambia cursos a modo G de 1 a 8
S Salta de un punto a otro
P Promoción de líneas curvas
D OVER 1 (sobre impresión)
N OVER 0 (normal)
F Fin del dibujo y acceso a cassette
Para las curvas hay que dar dirección y curvatura

30/1/1986 MICROCHESS 1K8
Se trata del juego de ajedrez adaptado para que ocupe menos de 16k8. El resto de instrucciones van por pantalla.
Todo un desafío para maestros!

REP. ORIENTAL DEL URUGUAY
CARMELO
ZARATE
CAMPANA
COLONIA DEL SACRAMENTO
BUENOS AIRES
RIO DE LA PLATA
LUJAN
MERCEDES
LA PLATA
CHIVILCOY
LOBOS
CNEL. BRANDSEN
SAN MIGUEL DEL MONTE
CHASCOMUS





SUPLEMENTO

DE APLICACIONES

La computodora nos ayuda a resolver muchos problemas y puede sustituirnos en diversos toreas, desde las más elementales de seguridod (prueba de eso es la alorma que logramas poner en marcha can lo Dreon Commadore 64), hosta escribir par nasotros (con los procesodares que incluimos para los CZ 1000/1500 y TK 83/85, y poro la C 64), o hablor con un vocabulorio ilimitado (presentamos un sintetizadar de vaz paro CZ 2000, TK 90X y TS 2068), y también pademos dar a la máquina un uso profesionol a través de periféricos (can lo interfase para CZ 1000/1500, TK 83/85).

CONSTRUYAMOS UNA ALARMA CON LA C-64



Esta vez, una sencilla computadora casera nos prestará un servicio para el cual quizá no haya sido pensada, pero necesario e interesante. Las posibilidades de una alarma controlada por computadora frente a los sistemas electrónicos usuales son muchas Por ejemplo se le puede dar prioridad a la apertura de una puerta en especial sonar la alarma cuando más de dos puertas hayan sido abiertas. o programer una secuencia, sólo conocida por el usuario, frente a la cual el circuito no actúe En este proyecto, utilizaremos el USER PORT de la C-64, el cual ee fácilmente accesible en la parte posterior de la misma Este port, nos permite transmitir o recibir datos en sene o paralelo, aunque en aste último modo con algunas restricciones. En le figura 2, vemos los circullos de la C-64 que controlan el USER PORT Estos son dos integrados tipo 6526 CIA, y pueden ser controlados por el usuario mediante pokes adecuados. En la figura 3 podemos ver una tabla en donde se resumen todos los registros del integrado U2 En pnmer lugar está la dirección del registro en decimal luego en hexadecimal, y finalmente la función del mismo En nuestro caso, nos concentraremos en los 4 primeros de la lista. Dado que si bien no es necesario conocer su funcionamiento para llevar a cabo este proyecto, consideramos que es interesante saberlo

Principio de funcionamiento

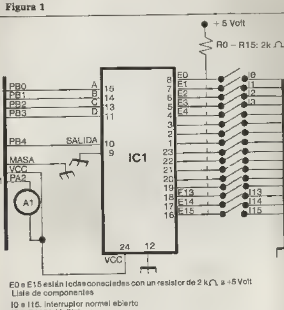
Como se puede ver, hay dos ports de entrada/salida: el port A y el port B.

Dado que son de igual funciona-

```
10 POKE 56579,15
15 POKE 56578,PEEK (56578) OR
4
20 FOR B=0 TO 15
30 POKE 56577,B
40 LET A=PEEK (56577) AND 16
45 IF A=16 AND B=15 THEN GO TO
90
50 IF A=16 THEN POKE 56575,PEE
K (56576) AND 251
55 IF A=16 THEN STOP
60 NEXT B
70 GO TO 20
90 FOR C=1 TO 50000: NEXT C
100 GO TO 50
```

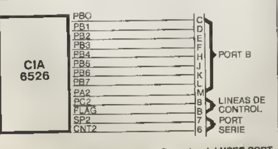
miento, nos referiremos sólo a uno de ellos, pero las explicaciones valen para ambos.
En nuestro caso, tendremos accesible el port B
Veamos el registro de dirección de datos DDRB (DATA DIRECTION REGISTER B).

Figura 1



E0 a E15 están todas conectadas con un resistor de 2 kΩ a +5 Volt
Lista de componentes
I0 a I15. Interruptor normal abierto
IC1 74150 Multiplexor
R0 a R15: Resistencias de 2 k ohm, 1/4 watt
A_1: Alarma

Figura 2



Conector del USER PORT

Como su nombre lo indica, mediante el DDRB podemos controlar la dirección de los datos en cada línea del port B, esto es decidir si la línea PB0 será una entrada o salida, lo mismo para el PB1, etcetera.
Asumiendo que el registro es de ocho bits, y que a cada bit le corresponde controlar una línea del port B, éste será salida si el bit correspondiente es 1 y entrada si es cero.
Por ejemplo, si ejecutamos POKE 56578,255, estamos poniendo todos los bits del registro a 1 y por lo tanto las líneas PB0 a PB7 serán salidas.
Si ahora hacemos POKE 56578,1 como podemos ver en el diagrama sólo ponemos un bit a 1, el correspondiente a PB, por lo tanto, éste (PB0) será salida y los restantes entradas.

```
P P P P P P P P
B B B B B B B B
7 6 5 4 3 2 1 0

0 0 0 0 0 0 0 1
```

Ya sabemos cómo hacer para que cada una de las líneas del port B sea una entrada o una salida
Ahora nos falta ver cómo hacer para poner datos en estas líneas.
Esto se lleva a cabo mediante el Registro de Datos Periféricos PRA (PERIPHAL DATA REGISTER A).
Entonces para poner un dato, digamos el valor 200 en el port B hacemos. POKE 56577,200;
asumiendo que todas las líneas de B fueron inicializadas como salidas.
Si quisiéramos leer un dato presente en el port B, hacemos:
PRINT PEEK 56577.
Al contrario del caso anterior, suponemos que el DDRB fue Pokeado con 0.
Para nuestra alarma, usaremos el port B para controlar un multiplexor.
Esto no es mas que una llave accionada electrónicamente que nos permite ir seleccionando cada uno de los sensores de alarma.
Por ejemplo, cuando en las entradas ABCD del IC1 (fig 1) tenemos el código binario 0001, estará seleccionando el sensor N1, y la computadora leerá el estado del mismo
Si en cambio, en las líneas ABCD tenemos el código 0101 (binario) que corresponde al 5 en decimal, la computadora leerá el estado del sensor número 5.
Entonces, si en las líneas ABCD ponemos todos los códigos desde el 0000 al 1111, iremos leyendo el estado de todos los sensores, desde el número 0 al 15, y determinar si existe una condición de alarma o no.
En la figura 1 podemos ver el esquema eléctrico completo de la alarma.
Por sensores utilizaremos inte-

...ptores de cualquier tipo, basta ... que se puedan adaptar a nues... necesidades, como ser colo... cados en al marco de una puerta y ...cionarse al abrirse la misma.

... una puerta es abierta, los con... tactos del interruptor correspon... dientes se cerrarán causando que aparezca un potancial de 0 volt en la entrada correspondienta de IC1

Mediante un sencillo programa, se hace un "barrido" de las 16 antra... das de IC1 genarando un contador binario an las líneas PB0 a PB3, que están aléctricamante conec... tadas a ABCD

Luego de cada conteo, la salida de ... terminal 10) es sensada le... yendo el astado de PB4

... en PB4 hay un potancial alto (equivale a al bit en 1) significa que ... alarma no ha sido disparade.

... lo contrario (el bit a 0) alguno de ... interruptores ha side activado y la alarma debe dispararse

Para ello, se genera un bajo potencial en PA2 causando de esta modo que suene la alarma.

Explicación dal programa

LINEA 10: pokea o escribe el valor 15 (00001111) en el registro de dirección de datos del port B. Esto hace que las líneas PB0 a PB3 sean salidas, y PB4 a PB7 antradas. Las líneas PB0 a PB3 son utilizadas para generar una cuenta an binario (de 0000 a 1111) en las líneas ABCD del multiplexor. Esta nos dará una salida an el terminal 10 qua representa el estado del interruptor seleccionado por el contador binario.

LINEA 15: pokea o escribe el registro de dirección de datos tan sólo cambiande al bit correspondiente al PA2 y manteniendo los otros bits inalterados.

Esto as necesario porque estas otras líneas controlan el bus serie y no deben ser modificadas.

LINEA 20: es el loop que genera la cuenta de 0 a 15. Esta seleccionará una de las 16 entradas del multiplexor cuando sea escrita en el registro de datos periféricos dal port B an la línea 30

LINEA 30: pokea o escribe la cuenta binaria an PRB.

LINEA 40: lee el PB y se ejecuta un AND 16 para que el único bit importante sea el bit 4, es decir la línea PB4 que representa al astado dal interruptor seleccionado. Si el valor leído es 16, esto indicaría que el interruptor correspondiente fue carrado, y la alarma debe activarse

LINEA 45: nos de un retardo en al disparo de la alarma en caso de qua la puerta abierta haya sido una escogida de antemano. Para nuestro ejemplo será la controlada por el interruptor número 15, pero esto puede modificarse cambiando el valor que se le asigna a la variable B. Por ejemplo, podemos poner el ratardo en el interruptor correspondiente a la puerta principal, por la cual nosotros entraremos a iremos a desconectar la alarma antes de que se active

LINEA 50: activa la alarma poniende PA2 en 0. Esto se deba hacar sin modificar las otras líneas del port A, de ahí que se recurra al AND 251.

LINEA 55: finaliza el programa si la alarma es activada

LINEA 60: incrementa el contador que selecciona la próxima entrade dal multiplexor.

LINEA 70: salta a la línea 20, luego de que la cuenta alcanza 15, reseteando al contador a 0 y comenzando el barrido nuavamante.

LINEA 90: genera un retardo para los casos de entrada normal. Este retardo pueda ser variado cambiando el valor del loop **K64**

Figura 3

DECIMAL	HEXA		REGISTRO
56576	DD00	PRA	Registro de datos periféricos A
[illegible]77	DD01	PRB	Registro de detos periféricos B
[illegible]78	DD02	DDRA	Registro de dirección de detos A
56579	DD03	DDRB	Registro de dirección de detos B
[illegible]80	DD04	TA	Registro del timar A (bajo)
56[illegible]	DD05	TA	Registro del timar A (alto)
56582	DD06	TB	Registro del timer B (bajo)
[illegible]	DD07	TB	Registro del timer B (alto)
56584	DD08	TOD	Reg. de hora diaria (1/10 seg.)
[illegible]85	DD09	TOD	Reg. da hora diaria (segundos)
[illegible]	DD0A	TOD	Reg de hora diana (minutos)
[illegible]	DD0B	TOD	Reg de hora diana (horas)
[illegible]	DD0C	SDR	Registro de dalos serie
56589	DD0D	ICR	Registro de control de Interrup.
[illegible]	DD0E	CRA	Registro de control A
[illegible]1	DD0F	CRB	Registro de control B

PROCESADOR DE TEXTOS



COMP CZ 1000/1500, TK 83/85
CONF IMPRESORA 16 K RAM

El programa que les ofrecemos a continuación nos permite conectara nuestra computadora una sencilla impresora térmica para utilizarla como procesador de textos. Entre las características del programa se cuentan las siguientes:

·Efectúa automáticamente justificación a derecha cuando una línea excede los 32 caracteres.

·Permite alterar el texto, definir un nuevo párrafo e indicar la finalización del mismo.

·Realiza una impresión automática del texto una vez que se llena la pantalla, hecho esto, lo borra y espera la introducción de más texto

·Para aquellos que no posean una impresora, podrán de todos modos correr este programa y verlo en funcionamiento mediante una sencilla modificación del mismo.

Comandos del sistema

Los comandos del sistema son pocos y sencillos de utilizar:

",": Indica el final de un párrafo. De este modo, automáticamente bajamos de renglón para tipear una nueva línea.

"/": Indica el final del texto. Al tipear este caracter ocurren las siguientes cosas: 1) Se realiza una impresión automática del texto existente hasta el momento.
2) Borra la pantalla y espera que más texto sea introducido.

"0": Entra al modo de edición de textos

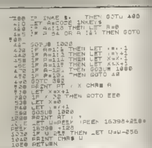
"**NEWLINE**": Equivale a espacio, y luego de alterar el texto, presionando SHIFT y N/L permite agregar más texto al ya impreso.

Modo de edición

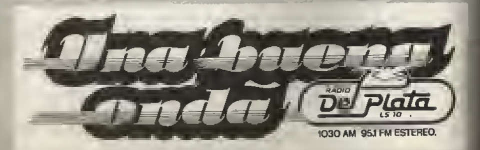
Al entrar al modo de edición, presionando 0, podrá verse la primera letra del texto en modo inverso. Mediante las teclas 5, 6, 7 y 8 (manteniéndolas apretadas) se podrá mover el cursor inverso en la dirección señalada por las flechas. Cuando el cursor esté sobre la letra que se desee borrar, basta con tipear la nueva letra sobre la anterior.

Para aquellos que no posean impresora, pero deseen ver cómo trabaja el programa deben cambiar las siguientes líneas.

330 PRINT AT 0,0;"COPIANDO"
331 INPUT I $
en vez de
330 COPY
sin línea 331 **K64**



```
   1 REM PROCESADOR DE TEXTOS
   5 DIM S(12)
  10 LET C$=
  20 LET L=0
  30 PRINT AT L,0;C$
  40 INPUT B$
  50 IF B$= .   THEN GOTO 275
  60 IF B$=     THEN GOTO 330
  70 IF B$= 0   THEN GOTO 300
  75 IF B$= /   THEN STOP
  80 LET C$=C$+B$+
  90 IF LEN C$ 32 THEN GOTO 30
  91 FOR I=32 TO 1 STEP -1
  92 IF C$(I)<    THEN NEXT I
 100 LET A$=(C$  TO I-1)+
                          )(1 T
0 32)
 110 LET C$=C$(I+1 TO )
 120 LET X=0
 130 FOR I=1 TO 32
 140 IF A$(I)          THEN GOTO 170
 150 LET X=X+1
 160 LET S(X)=1
 170 NEXT I
 180 IF A$(32)         THEN GOTO 30
0
 185 PRINT AT L,0;A$
 190 LET P=INT (RND*X+1
 195 IF S(A) 32 THEN GOTO 190
 200 FOR I=32 TO S(A)+1 STEP -1
 210 LET A$(I)=A$(I-1)
 220 NEXT I
 230 FOR I=A TO X
 240 LET S(I)=S(I)+1
 245 NEXT I
 250 GOTO 180
 275 LET C$=
 280 LET L=L+1
 290 IF L=21 THEN GOTO 330
 300 LET L=L+1
 310 IF L=21 THEN GOTO 30
 330 PRINT AT 0,0 [illegible]
 331 INPUT I$
 340 CLS
 350 GOTO 10
 360 LET Y=0
 370 LET X=0
 380 GOSUB 1000
 400 IF INKEY$    THEN GOTO 400
 410 LET A=CODE INKEY$
 420 IF A=118 THEN LET A=0
 425 IF A 34 OR A    THEN GOTO
 400
 427 GOSUB 1000
 430 IF A=112 THEN LET  =  -1
 440 IF A=113 THEN LET Y=Y+1
 450 IF P=111 THEN LET X=X-1
 460 IF A=114 THEN LET X=X+1
 470 IF A=121 THEN GOSUB 1000
 480 IF A=121 THEN GOTO 10
 490 GOTO 380
 520 PRINT AT   X CHR$ A
 510 LET X=X+1
 520 IF X 32 THEN GOTO 550
 530 LET X=0
 540 LET Y=Y+1
 560 GOSUB 1000
1000 PRINT AT  ;
1010 LET U=PEEK (PEEK 16398+256*
PEEK 16398 +128
1020 IF U 255 THEN LET U=U-256
1040 PRINT CHR$ U
1050 RETURN
```



CURSOS



Centro de Estudio de Disciplinas Informáticas

Carreras Cortas

ANALISTA PROGRAMADOR (1 años)
ANALISTA DE SISTEMAS (2 años)

Cursos teórico-prácticos regulares de:

Introducción a la computación, Diagramación lógica, Basic Básico Basic avanzado con manejo de archivos. Cobol, Logo para niños y adolescentes programación estructurada. Ejercitación práctica con problemas reales de nuestras empresas clientes. Cursos especiales para usuarios de commodore

CONSULTAS EXCLUSIVAMENTE AL 345-2351 DE 8 A 22 HS.

CURSOS - VARIOS NIVELES
DICTADOS POR PROFESIONALES

CON COMMODORE TK 90 SPECTRUM
CZ 1500 Y CZ 1000
VENTA DE COMPUTADORAS Y ACCESORIOS

CONTROSA — LAS HERAS 3281 SUBSUELO

---

micro cómputo

BASIC - LOGO

ASSEMBLER

MICRO COMPUTO - ACOYTE 44, LOCAL 8, TE.- 431-1081

COMPUTACION DE PRIMER NIVEL
SISTEMA AMERICANO

CURSOS

PARA EMPRESAS - PROFESIONALES - DOCENTES - ADULTOS - NIÑOS

LOGO - BASIC (I - II) - UTILITARIOS - GRAFICOS

HARDWARE - SOFTWARE - INTEGRADOS

AV. BORRECO 1196 CAPITAL — TEL. 864-0061

SOFTEEM COMPUTACION

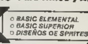
CURSOS MARZO ABRIL

BASIC LOGO - C/PM
PARA NIÑOS Y ADULTOS
PRACTICA C/COMPUTADORAS

INSCRIBASE = H. YRIGOYEN 1427 7º 8
38-7897 ESTACIONAMENTO GRATIS
H YRIGOYEN 1463

LE ALPI COMPUTACION

Cursos de Programación
BASIC Para Niños y Adultos



AV. CORRIENTES 4145 — INFORMES 88-7115

CLUB DE USUARIOS DE TI 99

CENTRO DE EDUCACION INFORMATICA

COMIENZAN LOS CURSOS

- BASIC	*- ASSEMBLER*
- LOGO	*- UTILITARIOS*

PUEYRREDON 860 0º P. TEL: 86-6430 / 89-4609

COSMOS

CENTRO INTEGRAL DE COMPUTACION

CURSOS BASIC/LOGO niños adolescentes adultos

Cursos para docentes	Clases individuales y grupales
Cursos intensivos y de apoyo	Asesoramiento para colegios
Docentes especializados	Jornadas de logo para docentes
Gabinete Psicológico	Grupos reducidos

La Pampa 2209 TE 781 7715

COMPUTACION

- *Cursos individuales y grupales*
- *Cursos para adolescentes y estudiantes*
- *Cursos para profesionales y empresas*
- *Prácticas en IBM PC, Commodore*
- *Trabajos Prácticos de la Facultad*

BASIC, FORTRAN, COBOL
Procesador de Palabra, Base de Datos, etc.

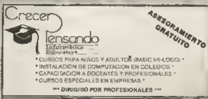
BECAS Y 1/2 BECAS

CPU - HUMAHUACA 4030 - CAPITAL
TE 86-0718 - LU a VI. de 17 a 21 HS

L & S SOFTWARE

- **Cursos computación**
- **Varios niveles**
- **Práctica en computadoras**
- **Vacantes limitadas**

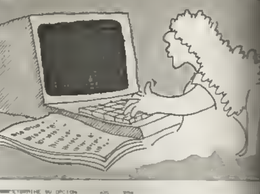
informes 431-8173 / 47-0260
11 de septiembre 2140 12 K CAP. FED.



PROCESA TEXTO

COMP. DREAN COMMODORE 64
CLAS. UTI

Este programa es un procesador de texto que posibilita utilizar la Commodore como máquina de escribir, y archivar en caset o imprimir con la impresora el texto deseado. Nos permite cargar un texto en una cinta, copiarlo por impresora, corregirlo.



```
1 [illegible]
2 POKE808,225
3 GOTO11
4 PRINT [illegible] FOR I=0TO50 GET A$ [illegible]
[illegible]
5 [illegible]
6 [illegible] PRINT [illegible] RETURN
7 [illegible] IF [illegible] <32 OR [illegible] 133 THEN [illegible]
8 [illegible]
9 [illegible]
[illegible]
15 [illegible]
19 PRINT [illegible]
20 PRINT "PROCESATEXTO"
21 PRINT [illegible]
22 PRINT [illegible]
23 PRINT [illegible] TRADUCIDO POR [illegible] D Y LEONE
24 FOR [illegible] NEXT [illegible] PRINT [illegible]
25 DIM T$(500) [illegible] 25
26 PRINT [illegible]
27 PRINT CHR$(147) [illegible]
28 PRINT PRINT [illegible] LO ESCRITO
19 PRINTCHR$ [illegible]
20 PRINT "PROCESATEXTO"
21 PRINT [illegible]
22 PRINT [illegible] LIPPERI
23 PRINT [illegible] TRADUCIDO POR [illegible] D Y LEONE
24 FOR [illegible] TO5000 NEXT [illegible] PRINT [illegible]
25 DIM T$(500) [illegible] 25
26 PRINTCHR$ [illegible]
27 PRINTCHR$(147) PRINTSPC [illegible] PRINT PRINT
28 PRINT PRINT [illegible] LO ESCRITO
29 PRINTSPC(15) [illegible] EDITAR LO ESCRITO
30 PRINTSPC(15) [illegible] AGREGAR ALGO
31 PRINTSPC(15) [illegible] CARGAR DE CINTA
32 PRINTSPC(15) [illegible] GRABAR EN CINTA
33 PRINTSPC(15) [illegible] TEXTO IMPRESO
34 PRINTSPC(15) [illegible] POSICIONAR TABULADORES
35 PRINTSPC(15) [illegible] MEMORIA DISPONIBLE
36 PRINTSPC(15) [illegible] VOLVER AL [illegible]
37 PRINT [illegible] TEXTO NUEVO
38 PRINT [illegible] SU OPCION [illegible]
39 IF [illegible] GOTO [illegible]
40 IF [illegible] GOSUB [illegible]
41 IF [illegible] GOTO [illegible]
42 IF [illegible] GOTO [illegible]
43 IF [illegible] GOTO [illegible]
44 IF [illegible] GOTO [illegible]
45 IF [illegible] GOTO [illegible]
46 IF [illegible] GOTO [illegible]
47 IF [illegible] GOTO [illegible]
48 IF [illegible] GOSUB [illegible]
49 [illegible] GOTO [illegible]
50 [illegible]
51 [illegible]
52 [illegible]
53 [illegible]
54 GOSUB [illegible]
55 PRINT [illegible] PULSE [illegible]
56 GET [illegible]
57 PRINT [illegible] RETURN
58 GOTO [illegible]
59 PRINT [illegible] VOLVER [illegible]
60 PRINT [illegible] BUSCAR UNA PALABRA [illegible]
61 PRINT [illegible] LINEA [illegible]
62 PRINT [illegible] PROXIMA [illegible]
63 PRINT [illegible] LA LINEA [illegible]
64 PRINT [illegible] INSERTAR NUEVA [illegible]
65 PRINT [illegible] CORREGIR LA LINEA [illegible]
66 PRINT [illegible] RETURN
67 [illegible]
68 [illegible]
69 GOSUB [illegible]
70 [illegible]
71 IF [illegible] PULSE UNA TECLA [illegible] GOSUB [illegible]
72 IF [illegible] GOTO27
73 [illegible]
74 [illegible] GOTO [illegible]
75 [illegible] GOTO [illegible]
76 IF [illegible] GOTO27
78 IF [illegible] GOTO [illegible]
79 IF [illegible] CHR$(13) [illegible] THEN [illegible]
80 IF [illegible] GOTO [illegible]
81 GOSUB [illegible] GOTO [illegible]
82 FOR [illegible]
83 GOTO [illegible]
84 PRINT [illegible]
85 [illegible] NEXT [illegible]
[illegible]
86 PRINT [illegible]
87 INPUT [illegible]
88 [illegible] GOSUB127
89 IF [illegible] GOTO127
90 [illegible] GOSUB [illegible]
91 [illegible]
[illegible]
105 PRINT [illegible]
[illegible] ESTE [illegible]
[illegible]
128 PRINT [illegible] MAQUINA DE ESCRIBIR [illegible] LOS MARGE[illegible]
129 PRINT [illegible] DE AMBOS LADOS [illegible]
130 PRINT [illegible] A ESCRIBIR [illegible]
131 PRINT [illegible] ESTE PROXIMO AL [illegible] DE LA LI[illegible]
132 PRINT [illegible] SE OIRA UN BEEP DE [illegible]
133 PRINT [illegible] TECLA DE BORRADO PUEDE [illegible] USADA [illegible]
134 PRINT [illegible]
135 PRINT [illegible]
136 PRINT [illegible] CONTRAR UNA LINEA [illegible] EL SIMBOLO
137 PRINT [illegible] CARACTER [illegible]
138 PRINT [illegible] DESEA [illegible] USE LA TE[illegible]
139 PRINT [illegible] COMO EL UNICO [illegible]
```

```
DE . INEA
PRINT "UNA VEZ QUE QUE HALLA FINALIZ
ADO SU
 TEXTO, PULSE -1' PARA VOLVER
 .
PRINT "UNA VEZ EN EL MISMO USTED POD
 DETERMI
PRINT "AR CUALQUIER TAREA A SEGUIR.

 INPUT "FIJE MARGEN IZQUIERDO.13-70/
 INPUTLM
 IF LM 70/ORLM<3)THENGOTO144
 PRINT "FIJE MARGEN DERECHO.13-70
 . IM=80-(RM+LM)
 IF RM 74 OR RM<3/THENGOTO146
 PRINT "ANTES DE COMENZAR " RETURN
 50 ... IE 4+5 9 POKES4+6 0 POKES

 ... POKES4,250 POKES4+4.33
 ... POKES4+2# 0
 ... POKES4 0 POKES4+# 80
 ...
FORIA=010,5 IFIW(TR/3)THEN156
 TTR /=0 RETURN
 ... TP=25 NEXT
 ...2TONI-2 Y$=/#+ Y=Y+1 PRINT
 ...
 ... X=32 RETURN

 ... THENK=K-5 PRINT "J" RETURN

 ...=500THENPRINT PRINT ...
 ... PULSE UNA TECLA
 ...=500THENGOSUB4 K=999 RETURN
 ...+1 GOTO160
 I=0
 PRINT"...

 PRINT "1. MARGEN IZQUIERDO"TAB(30)LM

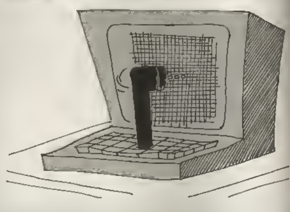
 PRINT"2. MARGEN AL TOPE"TAB(30)TP
 PRINT "3. MARGEN AL PIE"TAB(30)BM
 PRINT "4. LARGO DE LA PAGINA TAB(30

 PRINT "5. LINEAS ENTRE LINEAS TAB(30
)LS
171 PRINT "6. NUMERO DE PAGINA"TAB( ...
172 PRINT "7. NUMERO DE ...
TAB(30) PN
173 PRINT "8. CANTIDAD DE EJEMPLARES ...
TAB(30) CC
174 PRINT"PULSE -1 PARA PR ...
 IMPRIMIR 0
175 PRINT "UN NUMERO (1-8) PARA ...
 OPCION.
176 GOSUB4
177 IFX=133GOTO204
178 IFX>0 THENONXGOSUB180 183 186,189,1
92 195 198 201
179 GOTO165
180 PRINT "...MARGEN IZQUIERDO? <3-70/
 INPUTLM
181 IF(LM 70)OR(LM<3)THENGOTO180
182 RETURN
183 PRINT "...MARGEN AL TOPE? <5-60
 INPUTTP
184 IF(PL-TP BM 1 OR TP 5)THENGOTO183
185 RETURN
186 PRINT "...MARGEN AL PIE? <5-60
 INPUTBM
187 IF(PL-TM-BM<1)OR(BM<5)THENGOTO186
188 RETURN
189 PRINT "...LARGO DE LA PAGINA? <T-26
0 INPUTPL
190 IF(PL-TM-BM<1)OR(PL<1)THENGOTO189
191 RETURN
192 PRINT "...LINEAS ENTRE LINEAS? (1-4
) INPUTLS
193 IF(LS>4)OR(LS<1)THENGO ...
194 RETURN
195 PRINT "...NUMERO DE ...
 INPUTPN
196 IFPN<1THENGOTO195
197 RETURN
198 PRINT "...IMPRIME NUMERO DE PAGINA?
(S/N) GOSUB4
199 IF(X$="S")OR(X$="N") THENPN ... RETUR
N
200 GOTO198
201 PRINT "...CANTIDAD DE EJEMPLARES? <
1-... INPUTCC
202 IFCC<1THENGOTO201
203 RETURN
204 PRINT "...VERIFIQUE QUE SU IMPRESORA
 ESTE LISTA K=0 P3=PL-TM-BM P2=0
205 PRINT "Y QUE EL PAPEL ESTE ALINEADO.

206 PRINT ...
 GOSUB4 P2=PN OPEN4 4,7
207 FORI=1TOTM PRINT#4 NEXT
208 IFLEN(T$(K)) ... THENGOTO215
209 IFT$(K)= ... THENGOSUB220 GOSUB217 K=K
+1 GOTO207
210 IFT$(K) = E THENGOSUB221
211 PRINT#4 SPC(LM)T$(K) PI=PI+1
212 IFLS<>1THENFORI=2TOLS PRINT#4 PI=PI+
1 NEXT
213 IFPI>P3-1THENGOSUB217 GOTO207
214 K=K+1 GOTO208
215 PRINT#4 SPC(LM)SPC(INT(TM-LEN(T$
(K))/2)) ...
216 PI=PI+1 GOTO212
217 PRINT#4 IFPN<>0 THENPRINT#4, SPC(L
M)SPC(INT(IM-7)/2) ... P2 GOTO219
218 PRINT#4
219 P2=P2+1 FORI=3TOBM PRINT#4 NEX
T PI=0 RETURN
220 P5=PI FORPI=P5TOP3-1 PRINT#4 NEXT RE
TURN
221 GOSUB220 GOSUB217 C1=C1+1 IFC1<CCTHE
NCLOSE4 GOTO26
222 PI=0 P2=PN K=0 GOTO207
223 PRINT "...E PALABRA O FRASE DESEA
ENCONTRAR?" PRINT G=K
224 Y=0 Y$= ... GOSUB17 PRINT PRINT P
RINT ... Y$
225 K=K+1 IFT$(K)= E THENGOTO231
226 IFLEN(T$(K))<LEN(Y$)THENGOTO225
227 IFT$(K)=Y$THENRETURN
228 FORY=1TOLEN(T$(K))-LEN(Y$)+1
229 IFMID$(T$(K),Y,LEN(Y$))=Y$THENRETURN

230 NEXT GOTO225
231 PRINT ... PULSE UNA TE
CLA. GOSUB4 K=0 RETURN
```

INTERFASE DE CONTROL PARA CZ 1000/1500, TK 83/85



Casi toda persona que posee una microcomputadora habrá sido alguna vez interrogada acerca de qué cosas prácticas se pueden hacer con la misma.

Una de las aplicaciones más obvias es la de controlar dispositivos externos; pero, desgraciadamente, muy pocos microprocesadores o CPUs son diseñados para hacer esto directamente.

La mayoría de los fabricantes de microprocesadores ofrecen dispositivos (comúnmente circuitos integrados) llamados controladores de periféricos y están hechos para funcionar específicamente con una CPU. Estos controladores simplifican en gran forma la tarea de interfacear la microcomputadora con dispositivos periféricos, como ser disketeras, impresoras o modems. Afortunadamente, los ingenieros de Sinclair Research diseñaron su microcomputador en base al CPU Z-80 para el cual hay muy buenos controladores de periféricos. Uno de ellos, el Z-80 PIO, puede ser utilizado para construir una sencilla interfase paralelo para los microcomputadores del tipo ZX 81 (CZ 1000/1500, TK 83/85).

El Z-80 PIO

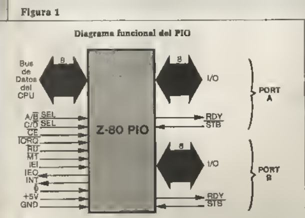
El Z-80 PIO (Pheripal Input Output) es un circuito integrado de 40 terminales diseñado para servir de interfase paralelo entre la CPU Z-80 y dispositivos periféricos. La comunicación entre el PIO y la CPU se produce conectando las líneas de datos del PIO directamente a las de la CPU. El PIO es un dispositivo de 2 ports. Esto significa que puede enviar y/o recibir dos sets de 8 bits de datos en paralelo.

Las líneas de control del PIO seleccionan uno de los dos ports (B/A SEL), habilitan el PIO (CE), y permiten que el PIO diferencie las palabras de control de las de datos (C/D SEL).

Las otras tres líneas de control (M1, IORQ, RD) aseguran un timing correcto durante las operaciones de entrada/salida con la CPU. Las barras sobre el nombre de las señales indican que su estado activo es el bajo.

Cada port tiene dos líneas de control (RDY y STB) que son utilizadas (no siempre) para sincronizar la entrada o salida de datos. En otras palabras, una de estas líneas, la RDY, debe ser activada para decirle a un dispositivo que está enviando datos al PIO algo así: "No manden más datos ahora. Todavía no estoy listo... O.K. ahora estoy listo. Manden datos ahora". El dispositivo debe responder activando la línea STB, diciendo: "O.K. acá van los datos. Tómelos ahora así puedo hacer otras cosas". Mediante el uso de estas dos líneas (lle-

Figura 1



Las máquinos diseñodos en bose ol microprocesodor Z-80 disponen de buenos controlodores de periféricos. Uno de ellos, el Z-80 PIO puede ser utillzodo poro el desorrollo que se explico en estos páginos.

vadas en inglés "hendsheke") la transmisión de datoe entre el dispositivo porifénco y el PIO se realiza en forma ordenede y eficiente.

El PIO contlene un número de registros internos que son utilizsdoe para controler eu oporación.

El mas importante es un registro mediante el cuel se pueden seleccionar diferentee modos de oporación para embos porta.

El PIO puede eer oparado en uno de cuatro modos, elendo aetos designados modo 0 a 3. Modo 0 es el modo de salide: las ocho lineas en el port designedo son utilizadas como salidas e algún dispositivo externo. En modo 1, el modo de entrada, todas las lineas del port serán entradas, reciblendo datos de algún dispositivo extemo. Modo 2 es el modo bidireccionel y está restringido al port A. En este modo, las líneae de control del port B, junto con las del port A, son utilizades para controler el flujo de datos en ambas direcciones del port A. Modo 3, el modo de control, es un híbrido entre el modo de entrade y el de salide; cuelquier linea del port especificado puede ser definida como entrada o salide.

El modo da control difiere del modo bidireccional en que une vez que una línea es designede como entrada o salide, se mantiene en esa condición y es necesario reprogramer para elterar le dirección de transferencie de datos en esa linea. Las líneas de control no son utilizadas en modo 3. Une explicación máe detalleda de los modos de oporación del PIO puede ser encontrada en el libro Z-80 PIO Z-80A PIO TECHMICAL MANUAL, Zilog, 1977.

Otros registros Internos del PIO son utilizados pera elmacenar el vector de interrupciones, una particularidad que distingue al Z-80 PIO. Ahora veremos como construir y programar une Interfase paralelo para las CZ 1000/1500 y TK 83/85 utilizando este PIO mediente control sin Interrupcionee ni "handsheke".

Construcción

La figure 2 nos muestra el circuito

eléctrico de la interfase paralelo. El port B es utilizado como entrada para leer el estado de ocho interruptores (I1-I8), mientras que el port A, es configurado como salida para manejar ocho diodos emisores de luz (LEDs). Se utilizan compuertas inversoras para amplificar la salida de cada línea del port de salida. La máxima corriente capaz de entregar el PIO es de alrededor de 1,5 miliamperes, lo cual no es suficiente para iluminar un led pero sí para manejar una entrada TTL, o cuatro entradas tipo LS TTL. Los pulsos de reloj invertidos se hallan presentes en el terminal 6B, del conector trasero de la computadora. Esta señal debe invertirse nuevamente antes de ser suministrada al PIO. Las líneas de control, STB y RDY de cada port, no son utilizadas. No son necesarias porque se seleccionará el modo 3 de trabajo. Dado que esta aplicación no necesita interrupciones, el terminal IEI (Interrupt Enable In) será conectado a +v, y el IEO (Interrupt Enable Out) no es conectado.

Las señales presentes en el conector trasero de la máquina pueden ser utilizadas mediante un conector de borde apropiado.

Este debe ser de 50 terminales, con un espacio entre contactos de 0,1 pulgada. Probablemente, el conector de borde deba ser modificado pues los que venden en los comercios tienen sus extremos cerrados, y nosotros necesitamos que estén abiertos.

Esto se puede llevar a cabo mediante un cuchillo filoso, o una pequeña sierra.

La interfase paralelo puede ser armada en algún tipo de plaqueta de armado sin soldadura. Esto nos permitirá solucionar cualquier inconveniente en forma rápida y no correremos el riesgo de arruinar un componente al soldarlo. Recomendamos poner capacitores de 0,1 microfaradio físicamente cerca de cada circuito integrado, entre las líneas de +V y masa, con el objeto de suprimir los picos que puedan aparecer en la fuente, así como las oscilaciones de alta frecuencia.

Una caja adecuada para la interfase puede ser comprada en cualquier comercio de electrónica. Debe ser lo suficientemente grande como para poder contener el circuito y la fuente de alimentación necesaria para la operación de esta interfase.

En cuanto a la fuente de alimentación, existen dos opciones.

Si usted planea agregar más circuitos en otro momento, le recomendamos que arme o compre una fuente de alimentación de 5 Volt, 3 Amperes. Un circuito de la misma se puede ver en la figura 3. El regulador de tensión nos asegura que a su salida tendremos siempre 5 Volts, fundamental para no dañar ningún circuito.

Por otra parte, si no planeamos agregar más circuitos al existente, todo lo que necesita es el regulador de 5 Volts, y tomar la tensión necesaria de la alimentación de la máquina. De este modo no podremos exceder un consumo de 300 miliamperes. K64

Continuará

Figura 2

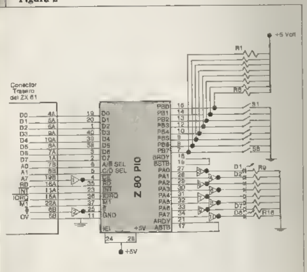
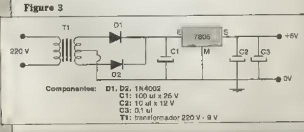


Figura 3



SINTETIZADOR DE VOZ

En este articula se explican las programas que permiten que la computadara "hable" can un vocabularia ilimitada, eliminanda así las dificultades que puede acarrear el almacenamienta de sanidos.

Ing. Pedro E. Colla



Un antarior articulo dedicado al elmacanamiento da sonidos, Incluyendo voz humana, permitia guardar en memona segmantos de conversación creando una harramlenta mediante la cual se puede hacer "hablar" al computador.

Sin ambargo, al mismo no callfica como un verdadaro sintetizador de voz puea el anganche de los segmentos de habla para formar palabras de un vocabulano más o menos grande as por cierto una tarea ardua.

En este articulo se proporciona un sintetizador de voz de vocabulario ilimitado, el cual no tiene los inconvenientes antes mencionados.

Tal como se había indicado an el articulo anterior, la voz humana se puede considerar an esencia formada por dos componentes; una señal de frecuencia más o menos alta y de amplitud constante mez-

ciade con otra de mucho menor frecuencie pero de emplitud variable.
Habiamos visto en esa oportunidad que, el sacrificar este último componente, introducia un grado moderado de distorsión pero sin destruir por completo la legibilidad.
Si segmentemos una conversación pasaremos de una frase a la palabra y luego a la sílaba, cuando segmentamos la sílaba nos encontramos con la unidad de la vocalización, el **fonema**.
Los fonemas son los bloques básicos con los que formamos nuestra conversación, e, intercambiándolos de distintas maneras, podemos formar nuestro casi ilimitado vocabulario a pesar que la cantidad de fonemas son relativamente muy pocos.
En efecto, basta un número algo superior a la cantidad de letras de nuestro abecedario para poder formar sin limite cualquier palabra.
La correspondencia entre letras del alfabeto y fonemas no es uno a uno pues hay letras que, de acuerdo al contexto en que se empleen, requieren la utilización de un fonema distinto; basta, a modo de ejemplo, considerar la utilización de la letra "C" en la palabra **casa** y en la palabra **cena** para comprender este particular.
Si aislamos cada fonema y observamos la forma de onda que tiene la señal de audio se verá que está formada por combinaciones de señales y silencios que se repiten incesantemente.
Un sintetizador de voz que utilize fonemas recurre a la representación de esas formas caracteristicas de señal y a su unión en rápida secuencia para formar palabras.
Debido al relativamente reducido número requerido se práctico el almacenamiento de éstos en la memoria de un computador.
El hecho de requerir formar una señal de audio partiendo de datos almacenados en memoria implica una velocidad de procesamiento que no se puede obtener de ninguna otra forma que no sea utilizando lenguaje de máquina, no importa que tan eficiente sea un intérprete o compilador de cualquier lenguaje de alto nivel (BASIC, PASCAL o FORTH) sencillamente no tendra la velocidad suficiente.
La señal de audio se forma entonces mediante la conmutación de la salida de MIC (y de parlante) de un computador SPECTRUM o TS2068 al ritmo impuesto por datos correspondientes a cada fonema bajo la supervisión de una rutina en lenguaje de máquina.
El programa se puede apreciar en la figura 1, en el mismo la seccion en lenguaje de máquina se almacena en instrucciones DATA para facilitar su carga y aquellos que no posean un programa ensamblador.
El programa BASIC almacena el código de máquina en grupos de diez bytes, colocando al comienzo de los mismos la dirección donde deberá cargar el primero de ellos y, al final, la suma de los mismos, de esta manera se facilita notablemente la depuración de eventuales errores de tipeo, comprensibles dado la exagerada extensión.
El programa sintetizador en sí mismo fue extraido de una publicación extranjera y sometido a pequeñas modificaciones francamente cosméticas, por parte del autor.
El mismo se aloja en la parte superior de la memoria RAM ocupando 2200 bytes a partir de la dirección 63000.
Esta ubicación no sa casual, le colocación en otra zona inferior implicaría que las interrupciones que reciba el procesador principal por parte del generador de video se harían perceptibles produciendo gangosidad en la salida del sintetizador.
El texto por sintetizar se coloca (pokes) en un buffer entre las direcciones 60000 y 62999, cas: media hora de conversación, finalizando él con un "0" hexadecimal (CHR #(0)); y sólo puede estar compuesto de letras mayúsculas y cualquier otro caracter (letras minúsculas, números o símbolos de puntuación) ocasionarán que no se prosiga con el remanente a partir de allí.
Esto representa cierta incomodidad en el uso práctico pues al margen de la limitación en cuanto el tipo de caracteres se requiere agregar una pequeña sección de programa BASIC que se encargue de armar el buffer para cada uso del sintetizador.
La solución viene de la mano de una corte rutine en lenguaje de máquina el cual funcione como "driver" para integrar el programa en la arquitectura del computador y permitir acceder al mismo a partir de la instrucción PRINT.
Para lograr esto se modifica el área de variable de sistemas denominada CHANS reemplazando las direcciones de manejo de dispositivos tipo "P" (Printer) por la dirección de comienzo del driver. Cada vez que se emplea LPRINT en vez de recurrir a la rutina en ROM destinada al efecto el caracter será enviado al driver.
Esta técnica fue profusamente explicada y empleada en un articulo anterior para el manejo de una im-

presora Centronics.
El driver se encarga de colocar el caracter en el buffer manejando la utilización de letras minúsculas y caracteres especiales; los primeros son transformados en mayúsculas y los segundos son reemplazados por espacios pues, de todas maneras, son impronunciables.
Los números son reemplazados por una secuencia de caracteres prevista en una tabla al efecto dentro del driver.

Cuando la sentencia LPRINT no concluye con ";" al último caracter enviado es un (cr) (carriage return CHR#(13)) al cual el ser recibido implica al llamado al sintetizador en sí mismo.
Si se utiliza ";" el programa asume que seguirán viniendo caracteres y maneja la situación "recordando" la próxima posición que tiene que utilizar en el buffer.
El driver también está incluido como sentencias DATA en el programa BASIC en la Fig. 1 y para referencia la expresión del mismo en Assembler puede visualizarse en la figura 2; aunque cabe remarcar que este último es incluído solamente en forma indicativa pues bastará con utilizar las sentencias DATA incluidas en la Fig. 1 para generar el segmento de código de máquina respectivo, el igual que el código del sintetizador en caso de haber problemas se exhibirá durante la carga del código el número de línea en que se encuentra la sentencia DATA con problemas.
Cuando logran cargar totalmente sintetizador y driver al mismo programa borre todas las sentencias DATA y las secciones de código destinadas a su carga en memoria quedando una versión reducida que se salva en cassette para uso futuro en la forma de un pequeño programa BASIC y dos segmentos "CODE" conteniendo respectivamente el sintetizador y el driver.
En la sección supervivienta de BASIC queda almacenada la información necesaria para modificar en cada carga el área CHANS de tal manera que al finalizar el sintetizador esté listo para su uso en forma aislada o como parte de otro programa.
Tanto el sintetizador como el driver son utilizables tanto para computadoras TS2068 como SPECTRUM, aunque para los usuarios de esta última máquina deberán tenerse en cuenta algunas modificaciones menores; estas son:
1) El área CHANS está ubicada en los computadores SPECTRUM ligeramente desplazada con respecto a la TS2068 y por lo tanto se deberán modificar las líneas 120 a 150 para reflejar esto, debiéndose tipear en lugar de lo indicado por la figura 1, lo siguiente:

```
120 POKE 23749,216
130 POKE 23750,214
140 POKE 23751,216
150 POKE 23752,214
```

2) El SPECTRUM no posee la sentencia DELETE para borrar instrucciones de un programa BASIC, por lo que no podrá ejecutar las líneas 525 a 546 de la Fig. 1, para obtener una versión reducida del sintetizador apta en el uso frecuente se deberá borrar "a mano" las líneas indicadas por las instrucciones DELETE y realizar los SAVE de líneas 550 a 570 no como parte del programa sino en forma manual.

Con las pequeñas precauciones derivadas de los puntos anteriores no existe otra dificultad en la utilización de este programa por igual en ambas máquinas.

Por último, nótese que, como parte del programa de carga, se hace decir al sintetizador algunos mensajes.
En los mismos se han utilizado deliberadamente errores de ortografía para mejorar la pronunciación; esto se debe a que solamente se maneja un fonema por cada caracter debiéndose utilizar con frecuencia este tipo de manejo para obtener una correcta pronunciación. En general será necesario reemplazar la "c" y "z" por "s", la "k" o "q" por "c" y la "v" por "b"; las palabras que empiezan con "h" deben ser escritas omitiéndola y cuando se requiera pronunciar el conjunto "qu" deberá utilizarse la "q" únicamente.
A pesar de estas precauciones se obtendrá pronunciación algo deficiente en determinados contextos para las letras "y", "j", "g", "l". Las palabras que utilicen "ch", "gu", "ll" o la letra "y" al comienzo no serán tampoco demasiado naturales.
Pese a las deficiencias apuntadas y al hecho que de todas maneras la pronunciación obtenida es e menudo difícil de interpretar el proyecto es interesante y obviamente de muy bajo costo.
Por otra parte al lector que se aventure al no poco tedioso tipeo necesario notará un efecto de acostumbramiento al cabo de cierto tiempo de uso que hará que el mismo suene cada vez más natural a medida que pase el tiempo de uso del mismo, incluyendo la detección de cierta "tonada" española que hace incontundible al origen del programa. **K64**

Aclaración: las líneas 525 a 546 por un error de impresión omiten el comando "DELETE" el cual debe ser agregado por los lectores.

PROGRAMAS

```
10 REM ------------------------
20 REM Speech
30 REM Sintetizador de Voz
40 REM ------------------------
50 REM Abre canal #4
60 R2M ------------------------
70 OPEN # 4,"P"
80 REM ------------------------
85 REM Genera direccion CHANS
87 REM para el driver el mismo
88 REM se coloca en 55000
90 REM ------------------------
100 POK2 23582,16
110 POK2 23583,0
120 POKE 26703,216
130 POKE 26704,214
140 POKE 26705,216
150 POKE 26706,214
152 GO TO 160
154 LOAD "" CODE 55000
156 LOAD "" CODE 63000
160 REM ------------------------
170 R2M Carga Sintetizador
180 REM ------------------------
185 CLS   CL2AR 54999
190 RESTORE 1000
195 PRINT  AT 0,0; FLASH 1;"Sintetizador —>'; FLASH 0
200 READ adress
210 IF adrese=99999 THEN  GO TO 340
215 LET t=0
220 FOR i=0 TO 14
230 READ byte
240 PRINT  AT 0,25;adreee+i
250 POKE adreee+i,byte
260 LET t=t+byte
270 NEXT i
280 READ sum
290 IF t <> sum THEN  PRINT 'ERROR EN LINEA '
    adress-52000  STOP
300 GO TO 200
310 REM ------------------------
320 REM Driver del Canal
330 REM ------------------------
340 RESTORE 5000
345 PRINT  AT 1 0  FLASH 1 'Sintetizador —>'  FLASH 0
350 READ adress
360 IF adress=99999 THEN  GO TO 500
365 PRINT  AT 1 25 adress+i
370 LET t=0
380 FOR i=0 TO 14
390 READ byte
400 POKE adress+  byte
410 LET t=t+byte
420 NEXT i
430 READ sum
440 IF t <> sum THEN  PRINT 'Error en Linea '
    adress-50000  STOP
450 GO TO 350
500 PRINT #0;'a finalizado la karga 0123456789'
520 PRINT  AT 3,0  FLASH 1;"Colocar en Cassette"; FLASH 0
522 GO TO 530
524 PAPER 7  INK 0  BORDER 0: CLS : PRINT  AT 0,0; FLASH
    1;"Sintetizador listo'
 FLASH 0: PRINT #0;"sintetizador listo"
525 10,525
530 160,100
535 1000 5000
540 190,530
545 150,152
546 522,522
550 SAVE "Speech" LINE 10
560 SAVE "Speech.DRV" CODE 55000,250
570 SAVE "Speech.SNT" CODE 63000,2200
```

```
4980 REM ------------------------
4990 REM CODIGO DEL DRIVER
4995 REM ------------------------
5000 DATA 55000,195,222,214,000,116,234,245,059,219,214,1717
5010 DATA 55010,254 255 032,006,042 220,214,000,024.003,1050
5020 DATA 55020,032,050 234 241,245,254 013,040,034,214,1404
5030 DATA 55030,040,056,040,241,245,214,058,056,049,241,1248
5200 DATA 55040,245,214,063 056,020,241,245,014,091,056,1455
5050 DATA 55050 025,241,245,214,007 056,016,214 123,056,1291
5060 DATA 55060 022,024,010 241,175,050,219,214,119,205,1279
5070 DATA 55070 024,246,201,241,002 032,119,035 024,042,1026
5080 DATA 55080,241,024,249,241,214,032,024,244,241,071,1501
5090 DATA 55090 229,033,093,215,126,104,040,011,126,254,1211
5100 DAL4 55100 255 240 003 035 024,240,035 024,241,035,0940
5110 DAT4 55110 235 225 026 254 255,040,005,119,019,035,1213
5120 DATA 55120 024 246 034 220 214,035,175,119,001,052,1170
5130 DATA 55130 219 214 201 049 005 070 079,032,255,050-1262
5140 DALA 55140 068 079 003 032 255,051,004,002,069,003,0066
5150 DATA 55150 032 255 082 075 065,065 084,002,079,032,0541
5160 DATA 55160 255 003 003 073 070,075,079,032,255,054,1037
5170 DAT4 55170 003 063 073,003 032,255,055,003,073,069,0073
5180 DATA 55180 064 065 232 255 056 073,067 072,079,032,0025
5190 DATA 55190 255 057 070 065 069,006,069,032,255,045,1034
5200 DAT4 55200 003 063 000 079 032,255,255,255,026,000,1110
5210 DATA 55210 000 000 000 000 000.000,000 000 000,000,0000
5220 DATA 99999
```

```
990 REM ------------------------
995 R2M CODIGO DEL SINTETIZADOR
997 RE4 ------------------------
1000 DA1A 63000,100,177,200,242,070,020,107,025,004,020,1001
1010 DATA 63010,000,085,070,100,107,000,000,002,025,000,0074
1020 DATA 63020 254 200,170,000,010,007,030,070 062,007,1310
1030 DATA 63030,043,254,200 176 004,010,047-005,007,122,1001
1040 DATA 63040,179,000,000,000,103,010,010,201,001,097,1004
1050 DATA 63050,000,000,000,000,000,000,000,000,001,000,0090
1060 DATA 63060,003,002,000,007,000,000,000,004,000,010,0114
1070 DATA 63070,004,025,000,041,000,000,000,000,004,020 0160
1080 DATA 63080,001,001,001,000,001,010,001,000,007,000,0060
1090 DATA 63090,002-004,000,010,001,002,001,000,000,001,0007
1100 DATA 63100 000 010 000,000,000,000,002,000,007,030-0199
1110 DAT4 63110 010,000-001,010,000,000,000,010,002 010 0070
1120 DATA 63120,002,010 001 011,000,020,002,011,010,000 0000
1130 DATA 63130,001,010,002,002,601,000,007,010,002,010,0001
1140 DATA 63140,002,004,001,000,002,000,000,010,000,000,0001
1150 DATA 63150,003,010,001,000,107,240,033,000,224,241,1300
1160 DAT4 63160,012,000,042,246,000,000,000,100,234,030,1110
1170 DAL4 63170,000,052,254,000,210,100,007,004,123,210,1000
1180 DA1A 63180,100 247,004,001,056,007,004,007,210,100,1000
1190 DALA 63190,247,014,000,107,229,014,000,205 020,000,1026
1200 DALA 63200,000,000 022,000,247,060 000,000,102,107,0060
1210 DAL4 63210,000 007 000,200,100,011,000,100,177,202,1441
1220 DAT4 63220,100 007 004,100,107,220,030,064,031,043,1202
1230 DATA 63230,124,101 032 201 024,223,000 247,000,240,1401
1240 DAT4 63240,100 250 049,000 190 247,000,253,100,204,1040
1250 DATA 63250,004,001 220 247 072 046 000,049,013-204,1727
1260 DAT4 63260,037 240 146 240,073,047,244,047,000 249,1020
1270 DATA 63270 101 240 000,247,211,202,237,247,100 245 2042
1280 DAL4 63280 000 240 094 261,079 251,100,200,013,007 1405
1290 DATA 63290 000,020 020 022,020 007,079,020,004,025 0290
1300 DAL4 63300,050 003 045,010,010,014,005,002,023,000 0207
1310 DAT4 63310,025 121 027,100,010,000,000,003,007,000 0023
1320 DATA 63320,000 000 009 002,007,000,000,001,010,000,0060
1330 DAT4 63330,012 000,007,000,000,000,017,002,007,004,0070
1340 DA1A 63340 000,000 012 001,021,005 007,000,000,002,0074
1350 DATA 63350,005 002 010,000 000,003,023,001,000,004 0060
1360 DATA 63360,012,001 007,000,001,007 251,004,000,005,0000
1370 DAT4 63370,010 003 013,000 010,000 000,001,000,004 0005
1380 DATA 63380,200,000,002 000 032,002,000,004,007 004 0002
1390 DATA 63390,007,000 000 000 000,000,014,005,000,007,0001
1400 DATA 63400,000,000.004,000 012,000,010,005,007-002-0006
1410 DATA 63410 019 002,025.002,000,001,032,000,000,000 0245
1420 DAL4 63420,042 004 000,000,254,246,201,014,011,004,0024
1430 DALA 63430 000 073,012 000,003,110,007,010,004,010,0201
1440 DALA 63440 010 013,022,014 000,010,007,032,000,011,0134
```

```
.000 DATA 63480,002,012,013,000,000,010,100,012,000,000,0390
.460 DATA 63460,001,000,012,000,100,012,000,010,009,010,0242
1420 DATA 00470,000,000,000,140,032,107,001,000,000,000,0371
.000 DATA 03400,255,000,255,000,200,000,255,000,000,000,1070
.400 DATA 03400,200,000,255,000,200,000,000,000,255,000,1073
1000 DATA 03500,005,000,255,000,200,000,200,000,255,000,1270
.410 DATA 63510,255,000,002,000,200,000,000,000,255,000,1070
.000 DATA 03520,101,024,000,000,020,000,012,038,064,000,0700
.000 DATA 63530,073,000,103,017,010,000,064,001,015,000,0400
.064 DATA 63040,106,032,041,001,017,000,000,000,000,200,0450
.000 DATA 03000,000,255,000,000,000,100,000,121,000,255,1250
.000 DATA 03540,000,219,000,010,000,000,000,070,036,153,0010
.270 DATA 03572,017,010,000,004,001,010,000,100,064,041,0300
.000 DATA 63500,000,000,000,024,007,000,140,064,100,001,0070
.000 DATA 03000,001,000,001,000,034,024,010,010,000,000,0400
.000 DATA 03000,001,000,007,020,000,170,032,001,064,255,0000
12.0 DATA 03010,000,007,021,020,000,032,000,175,032,000,0412
.250 DATA 02070,047,100,032,250,020,000,000,000,000,255,1064
.000 DATA 02630,000,005,000,000,001,072,000,000,000,013,0011
.000 DATA 03640,010,010,000,000,100,000,000,000,250,012,0095
.050 DATA 23600,130,012,000,014,000,000,034,000,020,000,0302
.000 DATA 00400,010,000,030,010,032,010,130,000,024,000,0004
.270 DATA 03270,032,012,026,000,000,000,000,000,004,000,0097
.000 DATA 03000,032,012,000,000,014,000,000,012,010,014,0102
.690 DATA 63000,020,000,100,000,034,000,000,010,100,013,0400
1700 DATA 63700,030,014,000,000,024,000,020,009,010,000,0234
1710 DATA 03710,020,010,020,012,102,004,000,009,000,213,0304
1700 DATA 63700,030,002,000,000,000,000,064,000,030,012,0202
1720 DATA 63730,004,000,014,002,000,013,010,010,034,000,0130
1740 DATA 40740,100,000,024,000,030,010,130,012,020,014,0400
.700 DATA 02750,000,000,034,000,020,000,010,000,020,010,0220
.760 DATA 23700,020,012,130,004,000,000,032,012,026,000,0234
.770 DATA 03770,000,000,000,000,064,000,072,012,000,000,0204
1700 DATA 03700,014,000,000,000,000,000,012,010,010,022,0100
.700 DATA 03700,000,120,000,004,000,020,010,130,013,000,0402
1000 DATA 03000,014,000,000,024,000,020,000,012,000,020,0024
.210 DATA 23010,010,034,012,132,004,020,000,032,012,026,0310
1250 DATA 03020,000,000,000,064,006,004,002,000,013,004,0200
1250 DATA 03000,000,014,001,064,000,000,255,000,255,000,0577
.064 DATA 03040,005,000,000,200,255,000,055,000,255,000,1275
.050 DATA 02250,252,000,255,000,255,000,255,000,255,000,1270
.000 DATA 02000,250,000,255,000,055,000,255,000,255,000,1275
.070 DATA 03070,250,000,255,000,250,001,064,020,040,012,0004
.000 DATA 63000,042,012,005,001,020,017,046,000,047,007,0210
.030 DATA 02000,045,011,233,007,000,000,020,002,041,014,0015
.000 DATA 63000,035,024,031,015,029,000,027,027,027,031,0071
2:0 DATA 63010,032,225,201,001,005,000,000,100,000,255,0009
.250 DATA 03920,000,203,001,033,000,130,000,020,000,003,0470
.920 DATA 03930,001,077,000,007,000,000,001,070,000,027,0001
.064 DATA 03940,000,104,001,020,000,010,002,002,000,015,0275
1000 DATA 63950,001,000,000,007,000,030,020,072,001,046,0153
.000 DATA 03064,000,012,000,024,000,021,000,000,003,032,0123
.970 DATA 02070,000,000,003,010,004,000,000,007,002,010,0146
.000 DATA 03900,000,005,003,017,002,021,001,001,000,022,0140
.000 DATA 03990,000,000,003,015,003,020,000,000,000,005,0000
2000 DATA 04000,000,003,002,030,002,012,001,014,001,000,0134
2010 DATA 04010,000,075,002,000,001,021,000,002,000,000,0310
2020 DATA 64020,000,004,002,073,007,000,001,641,002,234,0372
2030 DATA 64030,014,001,005,001,000,007,020,000,005,000,0029
2040 DATA 06440,000,004,001,160,013,004,062,002,014,001,0472
2050 DATA 00056,643,000,000,007,000,100,000,009,009,031,0327
2060 DATA 06400,010,015,002,010,002,017,000,017,002,042,0131
2070 DATA 06470,000,019,000,013,007,014,000,010,000,100,0040
2000 DATA 04000,000,030,007,030,002,010,000,017,034,010,0142
2090 DATA 64090,010,024,010,452,001,000,000,000,000,014,0125
2100 DATA 64100,032,010,064,010,000,093,000,001,000,000,0172
2110 DATA 04110,014,000,005,105,002,000,064,126,042,172,0000
2120 DATA 04100,000,107,039,005,007,005,001,020,002,064,0445
2130 DATA 64130,000,000,000,112,064,151,036,146,030,097,0000
2140 DATA 04140,034,000,000,002,022,020,064,001,006,001,0145
2150 DATA 64150,020,001,022,001,000,000,070,002,002,030,0140
2160 DATA 64100,012,001,021,001,250,001,027,001,002,000,0133
2170 DATA 64170,037,009,010,001,004,001,000,002,011,001,4000
2100 DATA 64100,017,001,000,001,005,001,064,007,031,001,0103
2100 DATA 04150,032,001,000,001,000,001,000,001,020,001,0100
2200 DATA 94200,002,001,005,001,010,001,021,001,064,001,0165
2210 DATA 06410,039,001,031,001,156,001,122,001,019,001,0400
2220 DATA 64220,036,001,000,001,136,001,021,000,000,001,0229
2230 DATA 64230,101,001,021,001,023,001,640,001,000,001,0206
2240 DATA 64240,000,001,007,001,000,002,039,001,004,000,0117
2250 DATA 64250,042,001,011,001,014,001,004,001,051,001,0127
2260 DATA 64200,020,002,006,002,003,001,014,002,000,001,0155
2270 DATA 64270,005,001,010,004,001,000,015,001,021,002,0149
2200 DATA 64200,252,002,020,000,000,000,094,000,040,001,0640
2290 DATA 94290,063,001,007,001,004,001,012,001,000,001,0171
2300 DATA 64302,000,001,013,001,004,001,040,001,014,001,0001
2310 DATA 64310,019,004,000,001,000,002,001,004,020,001,0065
2320 DATA 04320,062,001,019,001,000,002,072,001,110,005,0279
2250 DATA 64330,000,001,000,001,020,009,007,000,010,153,0305
2340 DATA 04340,012,003,030,001,003,010,000,100,010,000,0261
2330 DATA 04350,010,005,001,106,000,000,255,000,255,000,2722
2340 DATA 64300,255,000,255,000,255,000,255,000,255,000,1275
2270 DATA 64370,255,000,255,000,255,000,255,000,255,000,1275
2250 DATA 64300,255,000,255,002,255,000,305,000,055,000,1275
2302 DATA 04300,250,000,255,000,250,000,255,020,255,020,1075
2400 DATA 06400,255,000,255,020,255,001,005,000,064,012,0004
2410 DATA 04410,042,012,005,001,033,017,046,000,047,000,0216
2420 DATA 64425,045,011,233,007,000,000,032,022,041,014,0412
2430 DATA 64430,035,024,031,015,022,025,037,027,027,001,0371
2440 DATA 64440,020,000,007,004,000,002,000,002,007,000,0077
0450 DATA 04450,000,001,012,000,012,004,007,005,002,000,0065
2460 DATA 04400,017,000,007,004,000,003,012,001,022,003,0072
2470 DATA 24470,007,000,000,004,003,000,010,000,000,002,0050
2400 DATA 04480,023,001,000,000,012,001,007,004,021,003,0002
2490 DATA 64400,021,004,000,005,010,000,412,002,010,000,0020
2500 DATA 64500,004,001,000,004,000,003,000,004,022,003,0075
2510 DATA 04210,000,004,007,004,007,004,000,000,000,000,0055
2520 DATA 04220,014,002,000,007,000,004,005,000,010,000,0271
2530 DATA 24530,010,002,007,022,019,002,007,004,009,002,0007
2540 DATA 64540,000,002,007,000,000,001,012,002,012,004,0000
2550 DATA 64550,007,005,000,004,017,000,007,000,000,003,0002
2560 DATA 64540,912,001,023,002,007,000,000,005,006,002,0002
2570 DATA 04570,010,000,000,003,023,001,000,004,012,001,0042
2580 DATA 24500,007,000,021,003,031,004,000,005,010,000,0030
2590 DATA 04590,013,002,010,000,004,001,000,004,009,002,0054
2600 DATA 04600,000,004,032,002,005,000,007,004,007,004,0001
2610 DATA 24610,000,000,000,000,014,000,000,007,020,004,0002
2622 DATA 24620,000,004,010,044,010,003,007,005,010,003,0074
2630 DATA 64630,007,004,009,022,002,002,007,000,000,001,0051
2640 DATA 64640,012,002,013,004,007,005,200,000,017,000,0072
2650 DATA 04030,007,004,002,003,012,001,020,003,007,005,0074
2660 DATA 64640,000,005,005,000,010,000,000,002,023,001,0064
2270 DATA 04470,000,004,012,001,007,004,021,003,031,034,0092
2600 DATA 04600,009,005,010,003,013,002,010,000,004,001,0020
2690 DATA 64490,000,004,009,002,000,004,030,002,005,004,0003
2700 DATA 64700,007,004,007,000,000,004,000,020,014,003,0052
2710 DATA 24710,304,007,000,004,002,004,010,000,010,003,0047
2720 DATA 04720,007,003,019,001,065,000,000,255,000,255,0005
2730 DATA 04730,250,255,000,255,000,255,000,255,000,255,1275
2740 DATA 04740,020,255,000,255,000,255,000,255,000,255,1075
2750 DATA 24750,000,255,000,255,000,255,000,255,000,255,1275
2760 DATA 64760,000,255,000,055,002,255,000,032,002,024,1030
2770 DATA 64770,011,015,000,012,029,010,010,012,007,010,0121
2700 DATA 04700,912,010,007,130,003,021,009,012,005,132,0270
2720 DATA 04700,004,007,012,012,000,015,009,013,000,034,0102
2800 DATA 64000,010,014,001,000,000,001,004,026,010,033,0130
2810 DATA 04010,000,255,006,027,006,147,032,023,010,025,0206
2820 DATA 04820,007,004,012,142,054,001,001,000,000,030,0310
2830 DATA 04030,024,236,000,007,013,006,029,013,017,007,0640
2840 DATA 64040,007,120,001,004,000,140,053,170,012,004,0542
2850 DATA 64030,004,123,034,023,005,101,001,000,000,000,0441
2860 DATA 04002,100,000,030,000,003,001,020,001,120,001,0030
2870 DATA 24070,020,001,053,001,077,001,047,001,072,001,0203
2880 DATA 04000,027,001,104,021,030,001,019,000,032,002,0302
2890 DATA 24090,015,001,026,001,007,001,030,000,072,001,0164
2900 DATA 64250,040,001,013,000,024,001,031,001,020,000,0140
2910 DATA 64910,032,002,000,009,010,009,004,002,007,005,0100
2920 DATA 64920,919,001,005,003,017,002,029,001,061,000,0137
2930 DATA 64930,022,000,006,002,015,003,022,001,005,001,0070
2940 DATA 94940,000,001,053,001,030,000,019,001,014,001,0130
2950 DATA 94400,000,062,075,003,000,001,021,001,099,001,0250
2960 DATA 64960,066,002,000,000,073,007,006,001,041,002,0204
2970 DATA 64970,234,014,031,005,001,003,007,002,005,025,0327
2900 DATA 64900,000,064,004,001,166,013,264,042,255,000,0715
2990 DATA 64990,255,000,102,929,179,012,009,019,002,000,0000
3000 DATA 65000,002,013,075,039,156,019,003,015,004,015,0344
3010 DATA 95010,000,032,159,021,001,016,005,010,092,033,0452
3020 DATA 65020,153,914,004,013,007,004,002,020,094,033,0343
3030 DATA 65030,166,013,005,013,000,014,094,001,034,000,0366
3040 DATA 65040,005,001,002,027,014,003,001,030,003,255,0347
3050 DATA 65000,000,022,020,024,010,002,000,029,017,040,0312
3060 DATA 65060,000,255,000,057,022,024,005,000,023,002,0040
3070 DATA 65070,012,003,001,021,019,042,002,255,002,064,0019
3080 DATA 95080,021,027,007,090,919,026,022,000,001,255,0500
3090 DATA 95090,000,044,021,025,001,097,021,024,014,255,0502
3100 DATA 65100,000,094,022,137,022,027,014,255,000,000,0051
3110 DATA 65110,022,033,905,647,014,029,027,024,010,037,0256
3120 DATA 65100,001,000,001,037,015,037,000,000,001,037,0141
3130 DATA 65130,002,003,002,024,012,010,001,014,021,005,0004
3140 DATA 65140,001,005,000,011,001,004,002,003,061,014,0042
3150 DATA 65100,400,000,255,000,255,000,255,000,255,000,1020
3160 DATA 65160,255,000,255,000,255,000,255,010,200,100,1330
3170 DATA 65170,100,010,200,100,100,010,200,100,100,000,0520
3180 DATA 65100,000,000,000,000,000,000,000,002,000,002,0000
3190 DATA 65190,000,000,000,000,000,000,030,000,000,000,0200
3200 DATA 95200,000,000,000,000,000,000,000,000,000,000,3000
4000 DATA 99999
```

```
00010         ORG 55000
00020 ;-----------------------------
00030 ;Speech.DAV
00040 ;
00050 ;DRIVER DEL CANAL 04 PARA
00060 ;SINTETIZADOR DE VOZ
00070 ;
00080 ;-----------------------------
00090 BEGIN   JP MAIN
00100 FLAG    DEFB #00
00110 ADRESS  DEFB #00,#00
00120 ;-----------------------------
00130 ;CUERPO PRINCIPAL
00140 ;-----------------------------
00150 MAIN    PUSH AF
00160 ;-----------------------------
00170 ;VERIFICA SI ES EL PRIMERO
00180 ;-----------------------------
00190         LD A,(FLAG)
00200         CP #FF
00210         JR NZ,FIRST
00220 FOLLOW  LD HL,(ADRESS)
00230         NOP
00240         JR PUT
00250 FIRST   LD HL,60000
00260 ;-----------------------------
00270 ;CARGA EN EL BUFFER
00280 ;-----------------------------
00290 PUT     POP AF
00300         PUSH AF
00310 ;-----------------------------
00320 ;VERIFICA SI NO ES CR
00330 ;EN CUYO CASO LLAMA AL
00340 ;SINTETIZADOR Y ACOMODA
```

```
;f1z9 PARA EL PROX1MO USO
;COLOQUE SU PRIMER BYTE
;EN 60000
;--------------------------------
        CP 13
        JR Z,CR
;--------------------------------
;EN CASO DE NO SER cr
;DESVIA A DISTINTAS
;RUTINAS DE ACUERDO AL
;CARACTER QUE SE TRATE
;ESTOS PUEDEN SER
;
; MAYUSCULAS
. MINUSCULAS
; NUMEROS
;
;LOS CARACTERES DE
;CONTROL Y ESPECIALES SON
;REEMPLAZADOS POR SPACES
;LOS NUMEROS SON
;CONVERTIDOS MEDIANTE UNA
;TABLA
;--------------------------------
        SUB 40
        JR C,SPACE
        POP AF
        PUSH AF
        SUB 58
        JR C,NUMS
        POP AF
        PUSH AF
        SUB 65
        JR C,SPACE
        POP AF
        PUSH AF
        SUB 91
        JP C,LETS
        POP AF
        PUSH AF
        SUB 97
        JR C,SPACE
        SUB 123
        JR C,MINS
        JR SFACE
;--------------------------------
;RUTINA DE cr
;--------------------------------
CP      POP PF
        XOR A
        LD (FLAG),A
        LD (HL),A
        CPLL 63000
        RET
;--------------------------------
;RUTINP DE ESPACIOS
;--------------------------------
SPACE   POP AF
        LD A,"
SPAC1   LD (HL),A
        INC HL
        JR FIN
;--------------------------------
;RUTINA DE MAYUSCULAS
;--------------------------------
LETS    POP AF
        JR SPAC1
;--------------------------------
;RUTINA DE MINUSCULAS
;--------------------------------
MINS    POP AF
        SUB 32
        JR SPAC1
;--------------------------------
.RUTINA DE NUMEROS
.--------------------------------
NUMS    POP PF
        LD B,A
        PUSH HL
        LD HL,TABLA
NUM1    LD A,(HL)
        CP B
        JR Z,NUMZ
NUM2    LD A,(HL)
        CP #EF
        JP Z,FIND
        INC HL
        JR NUM2
FIND    INC HL
        JR NUM1
NUM2    INC HL
        EX DE,HL
        POP NL
NUMZ1   LD A,(DE)
        CP #FF
        JP Z,FIN
        LD (HL),A
        INC DE
        INC HL
        JR NUMZ1
;--------------------------------
;PUTINA DE FINALIZACION
;ESTA ES INVOCADA SOLO
;SI LA INSTRUCCION DE
,IMPRESION FINALIZA CON
; ";" EN CUYO CASO EL
;DRIVER ASUME QUE
;CONTINUARAN LOS
;CARACTERES
,--------------------------------
FIN       LD (ADRESS),HL
          INC HL
          XOR A
          LD (HL),A
          DEC A
          LD (FLAG),A
          RET
;--------------------------------
;TABLA DE CONVERSION DE
;NUMEROS
;--------------------------------
TABLA   DEFB "1
        DEFM /UNO /
        DEFB #FF,"2
        DEFM /DOS /
        DEFB #FF,"3
        DEFM /TRES /
        DEFB #FF,"4
        DEFM /KUATRO /
        DEFB #FF,'5
        DEFM /SINKO /
        DEFB #FF,"6
        DEFM /SEIS /
        DEFB #FF,"7
        DEFM /SIETE /
        DFFB #FF,"8
        DEFM /OCHO /
        DEFB #FF,"9
        DEFM /NUEVE /
        DEFB #FF,"0
        DEFM /SERO /
        DEFB #FF,#FF,#FF
FINAL   NOP

S
FINAL           = #D7A8
NUMZ1           = #D748
FIND            = #D742
NUM2            = #D73P
NUMZ            = #D745
NUM1            = #D736
TABLA           = #D75D
FIN             = #D752
SPAC1           = #D724
MINS            = #D728
LETS            = #D728
NUMS            = #D730
SPACE           = #D721
CR              = #D717
PUT             = #D6EE
FOLLOW          = #D6E6
FIRST           = #D6EC
ADRESS          = #D6DC
FLAG            = #D6D9
MAIN            = #D6DE
BEGIN           = #D6D0

;
Start of source = 32768
Length          = 02753
```

GUIA PRACTICA

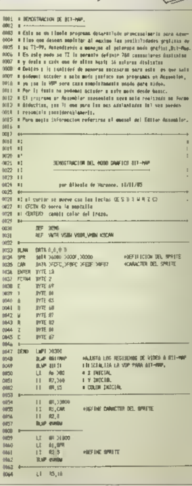
PROGRAMAS

DEMOSTRACION DE BIT-MAP



COMP. TI99/4A
CLAS. EDU

Utilicemos este programa en Assembler para graficar en nuestra TI, las instrucciones aparecerán por nuestra pantalla.



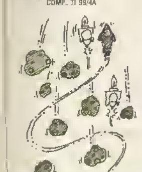
```
* DEMOSTRACION DE BIT-MAP.
* ------------------------
* Este es un simple programa desarrollado especialmente para aque-
* llos que desean explotar al maximo las posibilidades graficas de
* su TI-99. Aprendiendo a manejar el poderoso modo grafico Bit-Map.
* En este modo su TI le permite definir 768 caracteres distintos
* y darle a cada uno de ellos hasta 16 colores distintos
* Debido a la cantidad de memoria necesaria para esto  es que solo
* podemos acceder a este modo grafico con programas en Assembler,
* ya que la VDP sera casi completamente usada para video.
* Por lo tanto no podemos acceder a este modo desde basic.
* El programa en Assembler presentado aqui esta realizado en forma
* didactica, por lo que para los mas adelantados tal vez puedan
* resumirlo considerablemente.
* Para mayor informacion referirse al manual del Editor Assembler.
*
*-------------------------------------------------------------------
*:                                                                 *
*!                                                                 !
*'                                                                 :
*:          DEMOSTRACION DEL MODO GRAFICO BIT-MAP
!:          -------------------------------------                  :
!!                                                                 .
*!             por Alberto de Marenne. 18/11/85
*-------------------------------------------------------------------
*: el cursor se mueve con las teclas (E S D X W R Z C)             .
*! (FCTN 4) borra la pantalla
*! (ENTER)  cambia color del trazo.
*-------------------------------------------------------------------
       DEF  DEMO
       REF  VWTR VSBW VSBR,VMBW KSCAN
*-------------------------------------------------------------------
BLAN   DATA 0,0,0 0
SPR    DATA >4000 >000F,>0000            *DEFINICION DEL SPRITE
CAR    DATA >FCFC,>F8FC >FEDF >0F07      *CARACTER DEL SPRITE
ENTER  BYTE 13
FCTN4  BYTE 2
E      BYTE 69
X      BYTE 88
S      BYTE 83
D      BYTE 68
W      BYTE 87
R      BYTE 82
Z      BYTE 90
C      BYTE 67
*-------------------------------------------------------------------
DEMO   LWPI >8300
       BLWP @BITMAP         *AJUSTA LOS REGISTROS DE VIDEO A BIT-MAP
       BLWP @INIT           *INICIALIZA LA VDP PARA BIT-MAP.
       LI   R6 >80          * X INICIAL
       LI   R7,>60          * Y INICIAL
       LI   R8,15           * COLOR INICIAL
*-------------------------------------------------------------------
       LI   R0,>3800
       LI   R1,CAR          *DEFINE CARACTER DEL SPRITE
       LI   R2,8
       BLWP @VMBW
*-------------------
       LI   R0 >1B00
       LI   R1,SPR
       LI   R2 5            *DEFINE SPRITE
       BLWP @VMBW
*-------------------------------------------------------------------
       LI   R5,16
```

```
TECLA  CLR  @>837C
       CLR  @>8374
       BLWP @KSCAN
       LIMI 2
       LIMI 0
       JEQ  $
       JNE  TECLA
       AI   R5,16
       CB   @>8375 @E
       JNE  CX
       BL   @ARRIBA
       BLWP @PLOT
       JMP  TECLA
CX     CB   @>8375 @X
       JNE  CS
       BL   @ABAJO
       BLWP @PLOT
       JMP  TECLA
CS     CB   @>8375 @S
       JNE  CD
       BL   @IZQU
       BLWP @PLOT
       JMP  TECLA
CD     CB   @>8375 @D
       JNE  CW
       BL   @DERE
       BLWP @PLOT
       JMP  TECLA
CW     CB   @>8375 @W
       JNE  CR
       BL   @ARRIBA
       BL   @IZQU
       BLWP @PLOT
       JMP  TECLA
CR     CB   @>8375 @R
       JNE  CZ
       BL   @ARRIBA
       BL   @DERE
       BLWP @PLOT
       JMP  TECLA
CZ     CB   @>8375 @Z
       JNE  CC
       BL   @ABAJO
       BL   @IZQU
       BLWP @PLOT
       JMP  TECLA
CC     CB   @>8375 @C
       JNE  CCOLOR
       BL   @ABAJO
       BL   @DERE
       BLWP @PLOT
       JMP  TECLA
CCOLOR CB   @>8375 @ENTER
       JNE  CF4
       BL   @COLOR
       JMP  TECLA
CF4    CB   @>8375 @FCTN4
       JNE  TECLA
       BLWP @INIT
       JMP  TECLA
*------------------------------------
ABAJO  CI   R7,191
       JNE  AB1
       RT
```

```
AR1     INC  R7
        RT
*-----------------
MARIAD  CI   R7,0
        JNE  AR1
        RT
AR1     DEC  R0
        RT
*-----------------
IZQU    CI   R6,0
        JNE  IZI
        RT
IZI     DEC  R6
        RT
*-----------------
DERE    CI   R6,255
        JNE  DE1
        RT
DE1     INC  R6
        RT
*------------------------------------
COLOR   CI   R0,15
        JNE  CO1          BORRA?? COLOR DEL TRAZO
        LI   R0,1
        JMP  CO2
CO1     INC  R0
CO2     MOV  R0,R1
        SLA  R1,8
        LI   R0,>1003     *CAMBIO COLOR DE SPRITE
        BLWP @VWTR
        RT
*-----------------------------------------------
REG1    BSS  32
BITMAP  DATA REG1,BIT
*-----------------
REG2    BSS  32
INIT    DATA REG2,TO1
*-----------------
REG3    BSS  32
PLOT    DATA REG3,PLO
*-----------------------------------------------
BIT     LI   R0,>0040
        BLWP @VWTR        * AJUSTO VDP?? PARA BIT MAP
        LI   R0,>0206
        BLWP @VWTR        * CAMBIA TABLA DE PANTALLA A >1800
        LI   R0,>0403
        BLWP @VWTR        * CAMBIO TABLA DE DEF. DE CARAC. A >0000
        LI   R0,>03FF
        BLWP @VWTR        * CAMBIO TABLA DE COLORES A >2000
        LI   R0,>0701
        BLWP @VWTR        * COLOR DE PANTALLA NEGRO
        LI   R0,>0536
        BLWP @VWTR        * CAMBIO TABLA DE SPRITES A >1B00
        LI   R0,>0607
        BLWP @VWTR        * DEF.DE CARAC.DE SPRITES A >3800
        RTWP
*-----------------------------------------------
TO1     LI   R0,>1800
        CLR  R1
        LI   R2,0
R1      SWPB R1           * CREA LA TABLA DE PANTALLA
        BLWP @VSBW
        SWPB R1
        INC  R0
        INC  R1
        CI   R1,256
        JLT  R1
        CLR  R1
        DEC  R2
        JNE  R1
*-----------------------------------------------
        CLR  R0
        LI   R1,BLAN
        LI   R2,0         * BORRA TABLA DEFINICION DE CARACTERES
A2      BLWP @VMBW
        AI   R0,8
        CI   R0,>1800
        JLT  A2
*-----------------------------------------------
        LI   R0,>2000
A3      BLWP @VMBW        * BORRA TABLA DE COLORES
        AI   R0,8
        CI   R0,>3800
        JLT  A3
        RTWP
*-----------------------------------------------
PLO     MOV  @12(R13),R6  * TOMA X (R6)
        MOV  @10(R13),R7  * TOMA Y (R7)
        MOV  @16(R13),R0  * TOMA COLOR (R0)
        MOV  R7,R1
        SLA  R1,8
        LI   R0,>1B00
        BLWP @VSBW        MUEVE SPRITE
        MOV  R6,R1
        SLA  R1,8
        INC  R0
        BLWP @VSBW
        MOV  R7,R4
        SLA  R4,5
        SOC  R7,R0
        ANDI R4,>FF07
        MOV  R6,R5
        ANDI R5,7         * DEJA EN R5 EL BIT A PRENDER.
        A    R6,R4
        A    R5,R4
        MOV  R4,R0        * DEJA EN R0 EL BYTE A MODIFICAR.
*-----------------------------------------------
        BLWP @VWTR
        MOVB @MASC(R5),R3
        SOC  R3,R1
        BLWP @VSBW
        AI   R4,>2000
        MOV  R0,R1
        SLA  R1,12
        BLWP @VWTR
        RTWP
MASC    DATA >8040,>2010
        DATA >0804,>0201
        END
```

RUTA SOLAR

5º PREMIO DEL 3er CONCURSO K64
AUTOR: Fabio Rossi
COMP.: TI 99/4A





Tengo 16 años y escribí este juego para TI99/4A, con módulo de extended basic.

Consiste en evitar chocar con rocas, naves y monstruos que avanzan hacia nuestra nave. Una vez que pasamos estas dificultades llegamos a una base donde se cambiará la nave por otra de distinta definición. Cada base que pas... mos aumentará la velocidad de ... rocas.

Otro obstáculo es el combus... cuyo marcador se encuentra ... parte inferior izquierda de la pan... talla. Si éste marca vacio (agu... hacia la izquierda) la nave se ... a pique. Para evitar esto pode... llenarlo "chocando" con el tanque azul que pasa por el piso.

Además es imposible dejar la nave quieta, ya que si no el jugador ... dria buscar una posición en la cu... no chocaría nunca.

A la derecha del marcador de com... bustible hay una pequeña pan... que nos avisa la proximidad de ... naves enemigas y de las bases.

En el centro de la primer fila ... marca el record. Si éste es supe... rado se pasa a otra pantalla, en ... cual se le pide el nombre al juga... dor y se le toca la canción "Oh Su... sana".

Cuando aparezca la presenta... hay que apretar una tecla para ... cese el ruido de sirena y la com... tadora pida si desea instruccio... nes.

```
10 REM **************
20 REM **************
30 REM **RUTA SOLAR**
40 REM **************
50 REM **************
60 REM
70 REM
80 REM POR FABIO D. ROSSI
90 REM
100 REM NOVIEMBRE 1985
110 REM
120 REM PARA TI99/4A
130 REM
140 REM TI EXTENDED BASIC
150 REM
160 REM **PRESENTACION**
170 RECORD$="TI99/4A" :: RECORD=0 :: X=1 :: CALL CLEAR :: CALL SCREEN(2):: A=
180 CALL CHAR(40,"0I00IF003F007F008000F800FC00FE00FF007F003F000700FF00FE0
0")
190 CALL CHAR(44,"FF00FF00FF00FF000F000F000F000F00F000F000F000F000")
200 DISPLAY AT(6,4):"   (,,) ,  , ....  (,,)" :: DISPLAY AT(13,4):"(,,, (,,
 (,,) (,,)"
210 DISPLAY AT(7,4):"   , , ,  ,  -.  ,  ," :: DISPLAY AT(I4,4):".     ,
 ,  , ,  ,"
220 DISPLAY AT(8,4):"   ,,,+ ,  ,  -.  ....," :: DISPLAY AT(15,4):"*,,) ,
 ....  ...+"
230 DISPLAY AT(9,4):"   , . , .  -.  ,  ," :: DISPLAY AT(I6,4):"    , ,
 ,  , , ,-
240 DISPLAY AT(10,4):"   , ,, *,,+  -.  ,  ," :: DISPLAY AT(17,4):",,,+ *,,"
. .  . . ,," :: Z=800
250 CALL KEY(0,K,S):: GOSUB 290 :: CALL SOUND(-199,Z,5):: Z=Z+15 :: IF Z>1
EN 270 :: IF S=0 THEN 250
260 GOTO 310
270 CALL KEY(0,K,S):: GOSUB 290 :: CALL SOUND(-199,Z,5):: Z=Z-20 :: IF Z<7
N 250 :: IF S=0 THEN 270
280 GOTO 310
```

```
290 CALL COLOR(2,A,2):: A=A+1 :: IF A=17 THEN 300 :: RETURN
300 A=3 :: RETURN
310 FOR B=1 TO 8 :: CALL COLOR(B,A,2):: NEXT B :: CALL COLOR(12,A,2)
320 ME$="                              RUTA SOLAR POR FABIO ROSSI PARA TI99/4A
   DESEA INSTRUCCIONES (S/N)?                                   "
330 FIL=24 :: GOSUB 1700
340 CALL KEY(0,K,S):: IF S=0 THEN 340 :: IF K=83 THEN 360 :: IF K=78 THEN 470 EL
SE 340
350 FOR A=0 TO 30 STEP 7 :: CALL SOUND(-99,1000,A,-1,A):: NEXT A
360 ME$="                              UD DEBERA MANEJAR SU NAVE EVITANDO CHOCAR CO
N LAS ROCAS Y NAVES KAMIKASES                                   "
370 FIL=24 :: GOSUB 1700
380 ME$="                             EN EL BORDE INFERIOR IZQUIERDO DE SU PANTALLA
 ESTA EL MARCADOR DE COMBUSTIBLE                                 "
390 FIL=24 :: GOSUB 1700
400 ME$="                               PARA LLENARLO DEBERA CHOCAR EL TANQUE AZUL
QUE PASA POR EL PISO                                 "
410 FIL=24 :: GOSUB 1700
420 ME$="                               EN CADA NIVEL QUE AVANCE SE LE CAMBIARA LA
NAVE                            " :: FIL=24 :: GOSUB 1700
430 ME$="                               UD TIENE 5 NAVES / MANEJELAS CON LAS TECLAS
 E y X                           " :: FIL=24 :: GOSUB 1700
440 ME$="                               PRESIONE UNA TECLA PARA COMENZAR          BU
ENA SUERTE!       "
450 FIL=24 :: GOSUB 1700
460 CALL KEY(0,K,S):: IF S=0 THEN 460
470 FOR A=0 TO 30 STEP 7 :: CALL SOUND(-99,1000,A,-1,A):: NEXT A
480 NAVE=96 :: CALL MAGNIFY(3):: RESTORE 490 :: FOR HG=1 TO 10 :: READ BASE$(HG)
:: NEXT HG
490 DATA "LITIO","RUBIDIO","TITANIO","LANTANIO","ACTINIO","TANTALIO","CERIO","TO
```

```
R10","WOLFRAMIO","TECNECIO"
500 NIVEL=1 :: VIDA=5 :: T=0 :: NAVE=96 :: ENE=116 :: PUNTO=0 :: CALL CLEAR :: C
ALL SCREEN(2)
510 REM **PISO BASE**
520 CALL CHAR(86,"CCCC3333CCCC3333")
538 REM**COSO BASE**
540 CALL CHAR(87,"FF81BDA5A5B081FF")
550 REM **FONDO**
560 CALL CHAR(40,"")
570 REM **COMB**
580 CALL CHAR(88,"0303030303030303000000000000000000000000000000000"1
590 REM**EXPLOSION**
600 CALL CHAR(36,"0004001101014F0C002806240044200000000A200869704800A80220048720
8"1
610 REM **HOMBRE 1**
620 CALL CHAR(40,"1C1C1C1C0808182C2A290814167242030000000000000000")
630 REM **HOMBRE 2**
640 CALL CHAR(44,"1C1C1C1C080808181C1808080C744406000000000000000000"1
650 REM **HOMBRE 3**
660 CALL CHAR(60,"1C1C1C1C0808082C2A080C0A1C180C000000000000000000"1
670 REM **HOMBRE 4**
680 CALL CHAR(92,"1C1C1C1C08080818181C0A081C163223000000000000000000"1
690 REM **AVION 1**
700 CALL CHAR(96,"00000000000000000006060734FFE0F000000000000000000000000000C0FC1F
F8")
710 REM **PLATO 1**
720 CALL CHAR(104,"030303010101 0F0F0FFFE6667F3F1E07808080000080F0F0F0FF6766FEFCF
8E")
738 REM **PLATO 2**
740 CALL CHAR(108,"0000000000000001030707 0FFF7F3720000000000000000080C0E0E0F0FFFEE
C04")
750 REM **AVION 2**
760 CALL CHAR(100,"0000000000000000E0783CFE7F3FFF1F0000000000000000000000000080FEF
FC")
770 REM **NAVE**
780 CALL CHAR(112,"00000000071F080402FF3F0204081F0700000C0080F0204080FCFF804020F
080")
790 REM ***ROCA***
800 CALL CHAR(116,"0103061C3163464C40721A0A080F0100C0603C0 561434A66242460430E")
810 REM **FLECHA 1**
820 CALL CHAR(120,"00000000000007FC7C07000000000000000000000073CE08080E03C07"1
830 REM **FLECHA 2**
840 CALL CHAR(124,"00000000C1838710F71381 80C000000000000000003060E2CE78CEE2603")
850 REM **MONSTRUO**
960 CALL CHAR(128,"04041C147C3C6C0E0E0F070100000000000040607070787878 72FEFCFC7")
870 REM **TANQUE**
880 CALL CHAR(132,"0000031E3D303E1F0B0A1A163424200000000C078FCFC7CF8D05058682C240
4")
890 REM **ARBOL**
900 CALL CHAR(136,"E0BDE50771517121 3007E3A3E34F790100073DA7E0809C949C88E83800808
080"1
910 REM **EDIFICIO**
920 CALL CHAR(140,"FF9999FF9999FF9999FF9999FFFFC4C40000000000000000FFFF9999FFFF4
747")
938 REM ***COMIENZO***
940 CALL CLEAR :: CALL COLOR(2,16,16):: CALL SCREEN(21): FOR A=1 TO 18 :: CALL H
CHAR(A,1,40,321:: NEXT A
950 FOR A=3 TO 8 :: CALL COLOR(A,7,16):: NEXT A :: FOR A=21 TO 23 :: CALL HCHAR(
A,14,40,16):: NEXT A
960 PA$=STR$(RECORD)&"("&RECORD$ :: VOL=30 :: FIL=1 :: COL=10 :: GOSUB 1670
970 PA$="COMBUSTIBL" :: FIL=20 :: COL=1 :: VOL=30 :: GOSUB 1670 :: CALL HCHAR(21
,2,40,101
980 PA$="PUNTOS:" :: FIL=23 :: COL=1 :: VOL=30 :: GOSUF 1670 :: DISPLAY AT(24,1)
:PUNTO
990 CALL SPRITE(£2,NAVE,7,73,10,0,0)
1000 CALL SPRITE(£3,132,8,130,50,0,-10-NIVEL,£4,136,7,130,140,0,-10-NIVEL,£5,140
,12,130,170,0,-'J-NIVEL1
1010 CALL SPR TE(£1,88,7,160,821:: CALL SPRITE(£12,88,2,160,1):: CALL SPRITE(£13
```

```
88,2,160,97)
1020 FOR D=1 TO 2 :: PA$="ATENCION" :: FIL=22 :: VOL=0 :: COL=17 :: DOSUB 1670
1030 FOR A=1 TO 10 :: NEXT A
1040 PA$="((((((((" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 1670 :: FOR A=1 TO 10
 :: NEXT A :: NEXT D
1050 V=1 :: FOR A=6 TO 11 :: C=INT(RND*9)+3 :: VEL=(RND*10):: V=V+18 :: CALL SP
ITE(£A,ENE,C,V,240,0,-VEL-(N1VEL*1.5)-5):: NEXT A
1060 CALL KEY(0,X,S):: T=T+1 :: 1F T>300 THEN 1250 :: CALL MOTION(£1,0,-1):: CA
L DISTANCE(£2,£12,D):: 1F 0>22000 THEN 1120
1070 IF 0<1000 THEN 1150
1080 IF X=69 THEN 1090 :: IF X=88 THEN 1120 :: CALL CO1NC(ALL,C):: T=T+1 :: IF
=0 THEN 1060 ELSE 1150
1090 CALL MOTION(£2,-10,0):: CALL SOUND(-250,-5,25):: CALL MOTION(£1,0,0)
1100 CALL CO1NC(ALL,C):: T=T+1 :: 1F C=0 THEN 1060 ELSE 1150
1110 GOTO 1060
1120 CALL MOTION(£2,10,0):: CALL SOUND(-250,-6,25):: CALL MOT1ON(£1,0,0)
1130 CALL CO1NC(ALL,V):: IF V=0 THEN 1060 ELSE 1150
1140 GOTO 1060
1150 CALL D1STANCE(£12,£1,A):: IF A<18 THEN CALL MOTION(£2,14,0)ELSE 1170
1160 CALL DISTANCE(£2,£12,D):: IF 0<1500 THEN 1170 :: CALL SOUND(99,(0/100)+110
10):: GOTO 1160
1170 CALL OISTANCE(£2,£3,0):: IF 0<170 THEN 1200 :: CALL SOUND(1000,-7,2,110,20
:: CALL MOTION(£2,0,0)
1180 CALL PATTERN(£2,36):: V1DA=V1DA-1 :: FOR D=1 TO 10 :: CALL COLOR(£2,15)::
OR A=1 TO 10 :: NEXT A :: CALL COLOR(£2,7):: FOR A=1 TO 10
1190 NEXT A :: NEXT D :: FOR A=1 TO 100 :: NEXT A :: CALL OELSPR1TE(ALL):: IF V
DA=0 THEN 1720 :: GOTO 990
1200 A=200 :: CALL MOT1ON(£2,0,0,£3,0,0,£4,0,0,£5,0,0):: CALL MOT1ON(£1,0,6)
1210 CALL SOUND(-100,A,5):: A=A+5 :: CALL CO1NC(£1,£13,16,D):: 1F D=0 THEN 1210
:: CALL MOT1ON(£1,0,0):: CALL OELSPRITE(£2)
1220 CALL SPR1TE(£2,NAVE,7,116,10):: FOR A=0 TO 30 STEP 5 :: CALL SOUND(-99,100
,A,900,A):: NEXT A
1230 CALL MOT1ON(£3,0,-10):: CALL MOT1ON(£4,0,-10):: CALL MOT1ON(£5,0,-10):: CAL
L MOT1ON(£1,0,-1):: PUNTO=PUNTO+10
1240 OISPLAY AT(24,1):PUNTO :: GOTO 1060
1250 ENE=ENE+4 :: T=0 :: CALL MOT1ON(£2,0,0):: FOR A=6 TO 11 :: CALL OELSPR1TE(
A)
1260 FOR D=1 TO 20 STEP 7 :: CALL SOUND(99,(A*100)+300,D):: NEXT D :: NEXT A ::
CALL MOT1ON(£1,0,0):: FOR A=1 TO 500 :: NEXT A
1270 IF ENE=132 THEN 1280 :: PUNTO=PUNTO+100 :: O1SPLAY AT(24,1):PUNTO :: OOTO
010
1280 REM **BASE**
1290 ON NIVEL OOTO 1300,1310,1320,1330,1340,1350,1360,1370,1380,1390
1300 P=4 :: F=16 :: OOTO 1400
1310 P=5 :: F=6 :: DOTO 1400
1320 P=6 :: F=8 :: GOTO 1400
1330 P=9 :: F=16 :: GOTO 1400
1340 P=13 :: F=12 :: GOTO 1400
1350 P=13 :: F=16 :: GOTO 1400
1360 P=14 :: F=15 :: DOTO 1400
1370 P=15 :: F=5 :: GOTO 1400
1380 P=16 :: F=6 :: GOTO 1400
1390 P=16 :: F=15 :: GOTO 1400
1400 PA$="ATENCION" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 1670
1410 FOR A=1 TO 10 :: NEXT A
1420 PA$="((((((((" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 1670
1430 PA$="BASE("&BASE$(NIVEL):: VOL=0 :: FIL=22 :: COL=13 :: GOSUB 1670
1440 FOR A=1 TO 10 :: NEXT A
1450 PA$="((((((((((((((" :: VOL=0 :: FIL=22 :: COL=13 :: GOSUB 1670
1460 CALL MOTION(£3,0,0):: CALL MOTION(£4,0,0):: CALL MOTION(£5,0,0):: CALL MOT
ON(£2,0,30):: CALL SOUND(1500,-6,10)
1470 FOR A=1 TO 300 :: NEXT A
1480 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL SCREEN(F):: FOR A=1 TO 7 :: CALL
OLOR(A,P,F):: NEXT A :: CALL COLOR(8,P,P)
1490 B=10 :: C=12 :: CALL HCHAR(5,1,86,32):: FOR A=6 TO 10 :: B=B+1 :: C=C-2 ::
CALL HCHAR(A,B,87,C):: NEXT A :: CALL HCHAR(18,1,90,32)
1500 IF NAVE=112 THEN 1510 :: CALL SPRITE(£1,NAVE+4,7,122,172):: OOTO 1520
1510 CALL SPRITE(£1,96,7,122,172)
```

```
1520 CALL SPRITE(£2.NAVE,7,100,1,0,10):: CALL SOUND(I600,-6,20):: FOR A=1 TO 300
 :: NEXT A :: CALL SOUND(I300,-7,20):: CALL MOTION(£2,5,0)
1530 FOR A=1 TO 2I0 :: NEXT A :: CALL MOTION(£2,0,0):: FOR A=I TO 300 :: NEXT A
1540 FOR A=1 TO 10 :: CALL SPRITE(£3,92,I6,121,73):: CALL SOUND(10,2000,10,-3,10
)::: CALL COLOR(£3,2):: CALL SOUND(10,3000,10,-3,10):: NEXT A
1550 FOR A=1 TO 100 :: NEXT A :: CALL MOTION(£3,0,4):: FOR A=1 TO 13 :: CALL PAT
TERN(£3,40):: CALL COLOR(7,P,F):: FOR D=1 TO 20 :: NEXT D
1560 CALL PATTERN(£3,44):: CALL COLOR(7,F,P):: FOR D=I TO 20 :: NEXT D
1570 CALL PATTERN(£3,60):: CALL COLOR(7,P,F):: FOR D=1 TO 20 :: NEXT D
1580 CALL PATTERN(£3,92):: CALL COLOR(7,F,P):: FOR D=1 TO 20 :: NEXT D :: NEXT A
 :: FOR A=I TO 100 :: NEXT A
1590 CALL MOTION(£3,0,0):: FOR S=1 TO 100 :: NEXT S :: FOR A=1 TO 20 :: CALL COL
OR(£3,16):: CALL COLOR(£3,2):: NEXT A :: CALL DELSPRITE(£3)
1600 FOR A=0 TO 30 STEP 5 :: CALL SOUND(99,1700,A,-1,A):: NEXT A
1610 FOR A=1 TO 200 :: NEXT A :: FOR E=1 TO 10 :: PUNTO=PUNTO+50 :: DISPLAY AT(2
4,1):PUNTO :: CALL SOUND(25,1000,0):: CALL SOUND(25,1000,5):: NEXT E
1620 CALL SOUND(1300,-7,10):: CALL MOTION(£1,-5,0):: FOR A=1 TO 280 :: NEXT A
1630 CALL SOUND(2200,-6,10):: CALL MOTION(£1,0 10):: FOR A=1 TO 420 :: NEXT A ::
 CALL DELSPRITE(£1):: FOR A=1 TO 300 :: NEXT A
1640 IF NIVEL=10 THEN 1650 :: NIVEL=NIVEL+1
1650 NAVE=NAVE+4 :: IF NAVE=116 THEN NAVE=96 :: T=0
1660 ENE=116 :: CALL DELSPRITE(ALL):: CALL CLEAR :: CALL CHARSET :: GOTO 510
1670 REM **IMPRESION**
1680 FOR PA=1 TO LEN(PA$):: CALL HCHAR(FIL,COL+PA,ASC SE $(PA$,PA,1))):: CALL SO
UND(-100,(PA*20)+800,VOL):: NEXT PA
1690 RETURN
1700 REM***LETRADO***
1710 FOR ME=1 TO LEN(ME$)-27 :: DISPLAY AT(FIL,1):SEG$ ME$,ME,28):: FOR A=1 TO 1
0 :: NEXT A :: NEXT ME :: RETURN
1720 IF PUNTO<RECORD THEN 1920 :: CALL CLEAR :: CALL CHARSET :: CALL SCREEN(8)
1730 FOR A=1 TO 8 :: CALL COLOR(A,14,8):: NEXT A
1740 M$="81C3663C183C66C3" :: CALL CHAR(32,M$):: CALL CHAR(40,M$)
1750 PA$="FELICITACIONES''" :: FIL=3 :: VOL=30 :: COL=7 :: GOSUB 1670
1760 PA$="OBTUVISTE UN" :: FIL=6 :: COL=9 :: VOL=30 :: GOSUB 1670
1770 PA$="NUEVO RECORD" :: FIL=9 :: COL=9 :: VOL=30 :: GOSUB 1670
1780 PA$="ESCRIBE TU NOMBRE" :: FIL=12 :: COL=6 :: VOL=30 :: GOSUB 1670
1790 ACCEPT AT(20,7)SIZE(10):RECORD$ :: RECORD=PUNTO
1800 REM**CANCION**
1810 RESTORE 1820 :: FOR A=1 TO 57 :: READ DUR,NOTA :: CAL. SOUND(DUR,NOTA,5)::
NEXT A
1820 DATA 100,349,100,392,200,440,200,523,300,523,100,587,200,523,200,440,300,34
9,100,392
1830 DATA 200,440,200,440,200,392,200,349,400,392,200,392,100,349,100,392,200,44
0,200,523
1840 DATA 300,523,100,587,200,523,200,440,300,349,100,392,200,440,200,440,200,39
2,200,392
1850 DATA 800,349,400,494,400,494,200,587,400,587,200,587,200,523,200,523,200,44
0,200,349
1860 DATA 400,392,200,392,100,349,100,392,200,440,200,523,300,523,100,587,200,52
3,200,440
1870 DATA 300,349,I00,392,200,440,200,440,200,392,200,392,300,349
I880 PA$="DESEAS JUGAR DE NUEVO (S/N)?" :: FIL=24 :: COL=2 :: VOL=30 :: GOSUB 16
70
1890 CALL KEY(0,K,S):: IF S=0 THEN I890 :: IF K=83 THEN 1900 :: IF K=78 THEN 191
0
1900 CALL CLEAR :: CALL CHARSET :  GOTO 480
1910 CALL CLEAR :: PRINT "                CHAU ''" :: FOR A=I TO IO :: PRINT :: CALL S
OUND(400,233,10,262,20):: CALL SOUND(10,1000,30):: NEXT A :: END
1920 CALL CLEAR :: CALL CHARSET :: CALL SCREEN(12):: FOR A=1 TO 8 :: CALL COLOR(
A,I3,12):: NEXT A
1930 ME$="                              JUEGO TERMINADO              " :: FIL=10 :: GOSUB
I700
I940 ME$="                              DESEA JUGAR DE NUEVO (S/N)? " :: FIL=13 ::
GOSUB 1700
I950 CALL KEY(0,K,S):: IF S=0 THEN I950
I960 FOR A=0 TO 30 STEP 5 :: CALL SOUND(-99,1000,A,-1,A):: NEXT A :: IF K=83 THE
N 1970 :: IF K=78 THEN I910
1970 CALL CHARSET :: GOTO 480
```

SELECCIONAMOS

EL PROGRAMADOR DEL AÑO '86

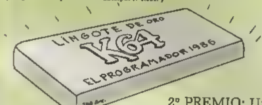
K64

COMPUTACION PARA TODOS

1º PREMIO

EL LINGOTE DE ORO DE K-64

(100 grs. Valor aprox. Bco Municipal A 1.000.-)



2º PREMIO: Una Consola 48 K

3º PREMIO: Una mesa para computador

BASES PARA PARTICIPAR EN EL CERTAMEN

Las bases y condiciones generales son las siguientes:

Una vez terminado y revisado tu programa, deberás enviarlo a la editorial grabado en un cassette o diskette, varias veces para mayor seguridad. (Inclusive grabado con dos grabadores distintos). Indicar en el cassette o diskette, los datos del programa, computadora y autor.

Otra condición es que sea original e inédito, es decir que no haya sido enviado a ninguna otra publicación. Si bien es preferible que vaya acompañado del listado del mismo por impresora, este no es imprescindible.

El programa deberá venir con un texto que aclare cuál es su nombre, objetivo, modo de uso, y explicación de cada una de sus partes, subrutinas y variables. Si posee lenguaje de máquina es fundamental una buena explicación sobre su funcionamiento e ingreso a la máquina. No olvidarse los datos completos del autor o autores.

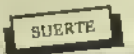
El texto se presentará en hojas tipo oficio y mecanografiado a doble espacio. No importa que la redacción no sea muy clara, eso queda por nuestra cuenta.

Jurado: Un jurado propio compuesto por profesionales en computación y usuarios de computadores decidirá los resultados del mismo.

El criterio de elección, como siempre, se basará en originalidad de la idea, método de programación; efectos gráficos y sonoros; documentación del programa; presentación y ahorro de memoria. En la clase de programas del tipo no-juegos, se evaluará también la facilidad de manejo y aplicación de los conceptos vertidos en el programa.

Cierre: El cierre de recepción de trabajos para concurso de programas será el: 31/07/86. (K64 se reserva el derecho de publicación de los programas recibidos, como asimismo la devolución del material)

Sorteo Mensual: Todos los meses se sortearán 20 cassettes entre los programas recibidos.



CRIPTOGRAFIA

COMP DREAN COMMODORE 64
CLAS ENT



Valiéndonos de este programa podremos efectuar y descifrar mensajes en clave, utilizando la misma técnica que empleó Alemania en la segunda guerra mundial. La única limitación que presenta es que no podremos incluir números, símbolos ni espacios.

```
1 REM 'CRIPTOGRAFIA' AU= ALEX BREED  A2
$= T.D.PINEIRO Y LEONE  BG#. RETURN
3 REM COPYRIGHT TCT 1980 INC CODE WORKS
4 REM BOX C505 SANTA BARBARA, CA 93160
5 REM AS OF 25JUL##
6 REM TRAD.& MOD.P.D.PINEIRO Y LEONE
7 REM PS DE W6OCTOS
20 REM ###VARIABLES UTILIZADAS###
21 REM CT=NRO.MAX.DE CIFRAS POR MECHSAJE
.#CI#T=ALMACENAMIENTO PJE,#DF#=INPUT RI

22 REM F=COD.ASCII D/CARACTER.EN CODIGO #
O=CONT.CEPACIOS.#L=C.O.CP CLAV CLAVE.
23 REM N=NRO.CIFRAS CII PJC,#PR#=TRPUT SE
CUENCIA FRACEC.#S#'/=OSI.COD.CP ASCII.
24 REM #TI=DET.CODIGO.CU#=FORMAIEU PANTP
LLA #W=RULO A/COD,#W#'/=SUB-COD EN ASCII
25 REM W/T=SUB-COD.R TADA COD #X(=DIMENC
IONES P/CODIGOS Y SUB-CODIGOS.
30 REM ###MODO DE TRABAJO D/PROGRAMA###
31 REM 100-230 DATA Y DETERMINA PROCODTA
IENICOS PARA DISTINTOS CODIGOS,
32 REM 200-260 INTCARGOA SOBRE PROG.AUTO
MATICA.C270-450 PREPARA MAQ.P USUARIO.
33 REM C60-500 MUESTRA CODIGOS USUARIO Y
 PIDE MENSAJE.#590 '60=INP.MJC.EN CODS
34 REM 770-900 PIDE OTRO MENSAJE CONT PIS
MA PROGRAMAC.#850-900 IMPR.CODS.Y S-CODS

35 REM 910-000 RUTINA SI 'Nu.#550-000 APL
ICA CIFRADO C/CARACTER.
60 GOTO 62000
100 CT=50 DIM CT# CT> XX(T3,26,
110 POKE VIC+24 23
120 DATA C,16,15,10.8.3,25,C,13.1 7.2C 2
0,10 12.9,1C,17 22,2,24,23,21 5,10,11
130 DATA 1.23,9 20 25,2 7,10 26 C TC,15
20.17,10,0,5 3,TC,22.16 12 21,11,5,15
140 DATA 1,0,2 20,3,1C,0.10,5,26 C 17 7
22,5,1C,10 23,11,21,12,25 13 15.10 20
180 DATA 1,3 2,5 C,10 C 1C,7,26,0,10,3 1
1,10,24 12 15 13,22 14,23,17,20 21,28
160 DATA 1,0,2,22 D 12 C,11 5.20 7 1C 9
24 0 13,10 17 10 25,10,26,10 23 10 21
170 X=RND(-1)/
171 PRINT "-----C W-ON ESTE PROGRAM
P USTED PD
172 PRINT "ORP EFECTUAR Y DESCIFRAR MENSA
JES EN CLA
173 PRINT "VE  USANDO LA MISMA TECNICA U
SADA POR C·
17C PRINT "ALEMANIA EN LA SEGUNDA GUERRA
MUNDIAL.
175 PRINT"                LA UNICA LIMITACI
ON DE ESTE
176 PRINT "SISTEMA RADICA EN LA IMPOSIBI
LIDAD DE CI·
177 PRINT "FRAR NUMEROS, SIMBOLOS Y ESPA
CIOS,
178 PRINT "           PARA COMENZAR PULSE CUA
LQUIER TECLA.
179 GET C# IF CC="" THEN GOTO 175
180 PRINT "     ' ...'
190 RESTORE
200 FOR W=1 TO 5
210 FOR K=1 TO 13 READ X,Y
220 XX(P XT=Y XX(P YT=X
230 NEXT K NEXT W
240 PRINT 77,
250 PR#= "PROGRAMACION #UIOMPIICA TC?N/
DFC= P  GOSUB 910
260 IF LEFT$(CT$, IT= 5 THEN GOSUB 010 GO
TO 460
270 T#= ABCDE
280 FOR I=1 TO 3
290 PRINT "CUAL DISCO EN RANURA I  \  T#
'T7    GOSUB 60000
300 IF LEN(IN#)<>1 THEN PRINT "UNA LETRA
POR FAVOR.  GOTO 290
310 IF IN#>="A AND IN#<="E THEN 330
320 PRINT "UNA LETRA DE LA "A" A LA "C",
 POR FAVOR   GOTO 290

330 T=ASC(IN#)-64
340 IF MID$(T I T=   THEN PRINT "DISC
O   IN#, YA ESTA OCUPADO.  GOTO 290
350 U#=    IF T>1 THEN U#=U#+LEFT$(T# T-1

360 T#=U#+    +MID$ T# T+1
370 HC T=INT N/ =T
380 NEXT I
390 FOR K=1 TO 3
400 PRINT "#CODIGO PARA DISCO K TA-Z)?
    
410 GOSUB 60000 IF LEN(IPC,<>1 THEN PRIN
T "UNA LETRA POR FAVOR.  GOTO 400
420 I=PSC IN#) 64
430 IF I T OR I>26 THEN PRINT "UNA LETRA
            , POR FAVOR! GOTO C00
440 = T=T S#(I T=I P#
450 NEXT K
460 PRINT "AQUI ESTAN LOS DISCOS Y CODI
GOS
470 PRINT "Y LOS NECESITA P/DESCIFRAR
EL MENSAJE
480 GOSUB 890
490 FOR K=1 TO 3 STK T=ASC(" T/-64 NEXT
K
500 PRINT "POR FAVOR DEPE SU MENSAJE.
510 PRINT "NO SE INGRESA UNA CIFRA EN BLANCO C
UANDO TERMINE.
520 FOR I=0 TO CT
530 PRINT "   #
540 GOSUB 60000 IF IN#="  THEN N=1-1 GOTO
590
545 IF IP#="0"THEN C0600
550 CT# I>=IN#
560 NEXT I
570 PRINT "ESO ES TODO LO QUE PUEDO MAN
IPULAR .. PERDONE.
580 N=CT
590 PRINT I IF N<8 THEN 770
600 I=0 D=0
610 PRINT "EL MENSAJE CIFRADO ES.....

620 FOR I=0 TO N CIC=CTCI T
630 FOR J=1 TO LEN(TCIC)
640 F=ASC(MID#(CT# J T),1)-64 P=P AND 127
650 IF F OR F>26 THEN 730
660 FOR W=-2 TO 7 L=3-ABS(T) GOSUB 950 N
EXT P REM L=1 2 3 2 1
670 PRINT CHR#(F+64)
680 K=K+T IF K<4 THEN PRINT   T K=0 0=0+
690 IF 0>5 THEN PRINT 0=0
700 S(1)=S(1)+1 IF S(1)<27 THEN 730
710 S(1)=1 S(2)=S(2)+1 IF S(2)< 27 THEN 7
30
720 S(2)=1 S(3)=S(3)+1 IF S(3)<27 THEN 7
30
730       3 =1
740 NEXT J NEXT I
750 PRINT IF FOR J=1 TO 5-K PRINT"X":
:NEXT:PRINT   NEXT
760 IF D< OR 1 THEN PRINT
770 PR#= "OTRO MENSAJE (S/N)" DF#= "S" GO
SUB 910
780 IF IN#= P THEN 60600
790 PR#= "OTRO PROGRAMA DE MAQUINA (S/P)
  DF#= S  GOSUB 910 IF IN#= "S"THEN490
800 GOTO 250
810 I#= "ABCDE" FOR K=1 TO 3
820 T=PT RND 1 =LEN(T#)+1
830 W(T=MID# T/ I/ N(K)=ASC(P(K)-6
C
840 U#=   IF T>1 THEN U#=LEFT#(T#,T-1)
850 T#=U#+MID#(T#,T+1
860 S(K)=INT(RND(1)*26)+1 S#(K)=CHR#(S(K
870 NEXT K RETURN
880 PRINT "ALRANURA    DISCO    CODIGO FPI
NT "
890 FOR I=1 TO 3 PRINT A PST(T S#(I) HEN
 PRINT " RETURN
910 PRINT TAB PR#"?    GOSUB 60000 IN#=L
EFT#(IN# 1) IF IN#="" THEN IN#=DF#
920 IF IN#= N THEN 60600
930 IF IN#= S  OR IN#= N THEN RETURN
940 PRINT "POR FAVOR RESPONDA  SI  O  NO

GOT 910
950 F=F+S(L)+T IF F<1 THEN F=F+26
960 F=X(T,W),F/
970 F=F+S(L)-1 IF F>26 THEN F=F-26
980 RETURN
60000 IN#=   ZT=1 ZC=2 ZO#=LAXT2UT
60010 GET Z# IF Z#<>  TPOI C0070
60020 IF ZT =1 THEN PRINT MID#(T  ZL,2C
 TI  ZC=2-: ZT=T1+15
60030 GOTO 60010
60040 Z=ASC(Z#) ZL=LEN(IN#) IF Z>12 AND Z<127
 THEN PRINT Z#   GOTO 60110
60090 IF ZL=0 THEN GOTO 60010
60100 IF Z=20 THEN PRINT Z#  Z#
60110 IF Z=13 THEN PRINT CR#, RETURN
60120 IF Z=20 AND ZL>0 THEN IN#=LEFT#(IN
# ZL-1  PRINT Z# GOTO 60010
60130 IF Z=141 THEN Z#=CHR#(-20#(ZL>0)) T
FOR T=1 TO ZL PRINT Z# NEXT GOTO 60000
60190 GOTO 60010
60200 ZT=TI/30 ZT=1 Z#=2 PRINT LEFT$(JC
# I  PRE  T
60210 IF PEEK(1ST AND 16)=0 THEN Z#=2 GO
SUB 60000 PRINT RETURN
60215 GOSUB 60000
60220 IF TI =ZT THEN GOSUB 60250 ZT=TI+1
5 Z#=3-Z#
60230 Z=PEEK(J5)AND 12 IF Z=T/ THEP Z3=
0 GOTO 60210
60240 IF T>Z1 THEN 60210
60250 IF Z=4 AND IN(JN THEN IP=IN+1 GOTO
60200
60260 IF Z=0 AND IN>1 THEN IN=IN-T GOTO
60200
60270 GOTO 60210
60280 PRINT (IN(+TT=IN)+ IN(+13)+MID#(JC#,2
S I) KIO# PRE JT+JW#(IN+TT+1,JNT, T
60290 PRINT #
60500 GET Z# IF Z#<> "0 THEN RETURN
60600 GET Z# IF Z#<> "THEN 60600
60605 POKE VIC+20 21 POKE VIC+21,0 PRINT
CHR#(5)
60610 GOSUB 61000 POKE VIC+33 C POKE VIC
+32,1C POKE SID+2C,0 PRINT "" END
61000 CR1=1024 VIC=53280 HD+4 CR#=CHR#(
13) SID=54272 JS=56320 CM=55296
61010 TP#= TI  GL=214 DI=235 RETURN
62000 LB=LEN POS(+2 CT=LEN(MUST+2 L2=LEN
(A2 T>= IF LI L THEN LI=L2
62010 IF L0CL)=2 THEN LM=L1+2
62020 B0#=LEFT#T ##
                    L0+2T
62030 DEF FN TIN=(#0-P/2 #10= T#/PIDAT
B0# 2 CT#1 TI#FN T/LIT T0#FP T/LIT
62035 GOSUB 61000 POKE VIC+32 0 POKE VIC
+33 0 POKE STO+20 0
62040 PRINT "    CHR#(CT POR I=1 TO C P
RIPT TMRTT0  B0# NEXT I
62050 PRINT "   TAD/FN TILEN(POST)/IPB
# PRINT
62060 FOR I=1 TO 3 PRINT IPC(T)I 01C NEX
T I
62070 PRINT "THE TROTFN TILEN(AUC T3,AUO

62080 IF R20    THEN PRINT B  TP0/FN TCL
CIPAZCTT P2# PRINT IP#/TIT 910
62090 B0#= PULSE  #RET#  P/ENTRAR  TR=PH
 TILEPTB0C(+27
62100 LM#=LEFT#(           LEN(B0#)T
62100 LN#=LEFT#(    LEP(B54
62110 PRINT TAB T02/ 0-       LP#   ;NNT
 
62120 PRINT TAB/10)     LM#  +
62140 PRINT TAB 3T    COPYRIGHT (C) T
985 TPC CODE WORKS.
62150 LI=LEFT 05#/ I=1 PRINT "ITTTTTT"
62160 PRINT SPC(T0/13),PTUN  1;TILEF
T#/B0# LIT, "T
62170 LI=LTI+1 IF LI<LEN(B0#)THEN LI=1 I=
3-I
62180 GET T#I IF T#="" THEN 62200
62190 IF PEEK(#STMND 16 THEN C2180
62200 CLP GOSUB C1000 PRINT "4 GOTO 100
```

D E B U G G I N G

SIMULADOR DE VUELO

Les escribo para comentar mi experiencia en pasar el programa SIMULADOR DE VUELO del Nº 3 a un cassette, sabiendo que así colaboro con una revista que valoro mucho y con los lectores que, como yo, deben haber sufrido bastante e la par que pensádamos mal do ustedes injustamente.

Los problemas con que me encontré fueron de 3 tipos: 1) tenía los ingredientes de la torta y no sabía cómo mezclarlos ni cocinarlos, siendo mi primer experiencia en lenguaje máquina; 2) errores impresos en el programa generador de la línea REM con 2926 caracteres, además de una tijera mal implementada en el programa en BASIC; 3) errores de mi propia cosecha en el tecleado, algunos por simple torpeza y otros por no haber distinguido bien ciertos detalles.

Mi esfuerzo terminó bien y ahora disfruto sin inconvenientes el programa, por lo que quiero compartirlo.

Me aboqué a generar la línea REM y descubrí que en ese programa debería decir 2 LET A = PEEK 16396 + 256 * PEEK 16396-2, en cambio, donde está subrayado dice 255. Pude darme cuente comparándolo con el gráfico de cómo queda la línea REM con 2926 caracteres (que en realidad corresponde a uno con 3000). También es diferente la línea 3 de como está corregida en el Debugging del Nº 6, porque el espacio en blanco está al principio y no al final (no sé si esto modifica el resultado). En dicho Debugging vuelven a equivocarse al intentar aclarar la línea 11, repitiendo 1,61 donde debe decir I,61.

Una vez que se tiene la línea 1 REM se agregan las líneas del cargador en lenguaje máquina, cuyo listado empieza con la posición 16514 y sigue con la 16515 inmediatamente al lado, cada renglón tiene 8 números que corresponden a 8 posiciones distintas y consecutivas (lo aclaro porque yo lo miraba sin entenderlo). En la pantalla sólo se podrá ver la línea 1 REM hasta el renglón 19 que termina con varios "?". Luego se agrega, empezando por la línea 2 el programa en BASIC. En este último, en la línea 8010, la tijera se comió renglones de espacio en blanco, siendo el objetivo formar 6 líneas en blanco.

Respecto a las cosas donde me equivoqué, las cuento porque pueden ser errores comunes a otros lectores. En la línea 7090 y 7091 se debe formar un cuadrado de 1 1/2 x 1 1/2 espacios, y no de 1 1/2 x 1. Además en las posiciones para pokear 17530 y 19176 van 33 y 203 respectivamente En



la línea 9074 del BASIC la variable deberá valer −8 (en mi ejemplar está borroso).

Finalizando, para controlar los pokeados conviene agregar estas líneas:

```
9901 INPUT B
9902 FOR A=ØTO 19440
STEP 8
9903 SCROLL
9904 FAST
9905 PRINT PEEK A;
TAB 4; PEEK (A+1); TAB
8, PEEK (A+2); TAB 12;
PEEK (A+3); TAB 16;
PEEK (A+4); TAB 20;
PEEK (A+5); TAB 24;
PEEK (A+6); TAB 28;
PEEK (A+7)
9906 SLOW
9907 IF INKEY$ = ""
THEN GOTO 9907
9908 NEXT A
9909 STOP
```

donde b será la primer posición a peekear. Para que coincida con el listado conviene introducir algunos de las mencionadas en la 1er. columna, pudiendo detenerse el programa con BREAK, continuar con GOTO 9903, y con GOTO 9901 comenzar y recomenzar. La línea 9907 es para regular el tiempo de aparición de los datos.

Dos detalles, la rutina "CRASH" comienza en 19369 y no en 19364, y el otro es que sería conveniente, para mejor entender los gráficos que incluyan una plantilla transparente cuadriculada.

Suerte a Uds. con la revista que es magnífica, y a los lectores con éste y otros programas.

M. Lewin

LA COMPUTADORA VE POR NOSOTROS

En el número 11, de febrero pasado, en la página 39, listado en código de máquina que es común a todos los programas de la serie presentada por Enrique González, le faltan estos bytes que se "cayeron" del pie de la página. Estos completan un total de 2180 bytes, o sea, de la dirección 63000 hasta la 65180 inclusive.

1	37	2	3	2	34	12	10
1	14	1	5	1	5	0	11
1	4	2	3	1	14	0	0
255	0	255	0	255	0	255	0
255	0	255	0	255	0	255	10
200	100	100	10	200	100	100	10
200	100	100	0	0	0	0	0

EL MERCADO ARGENTINO DE LAS HOME COMPUTERS

Continuamos con nuestra investigación sobre los productos disponibles en nuestro país y sobre las aplicaciones que le dan los usuarios.



"Como una herramienta"
Ing. Jorge KATZ (OMIS)

La home computer es una máquina muy versátil. No es necesariamente, como está implantada en la Argentina, para jugar. Es mucho más potente y se le puede sacar mucho más provecho. Esta fue una de las causas por la que nos abocamos a hacer un tipo de programas que la implemente como una herramienta en una pequeña o mediana empresa.

Actualmente, el concepto de home computer, de parte de la gente que está interesada en computación, es el de un aparato para jugar. La gente va a comprar el "juguete" para los chicos, y el padre dice "No... yo no sé nada. No me explique nada porque el chico es el que sabe". Entonces se le vende el teclado, la unidad de almacenamiento, una diskettera o grabador, utiliza su televisor y el joystick.

Eso forma una unidad de venta que los chicos usan para jugar. Los juegos los consiguen en muchos negocios o por intercambio. Y así se pasan las horas.

Considero que no es la finalidad última de la máquina. Si bien puede resultar un entretenimiento en determinadas circunstancias, el objetivo principal de la máquina es poderle sacar provecho. Utilizarla como una herramienta. La computación no es un fin, es un medio. Es como una máquina de escribir o de calcular, que le permite a la persona hacer sus tareas en forma mucho más eficiente, con menos trabajo y mejores resultados y productividad.

¿Qué significa utilizarla como herramienta? Si un señor tiene una ciendo a mano el listado de recibo de jornales de los empleados, la máquina, preparando un programa de sueldos y jornales, le permite hacer lo mismo automáticamente.

La ventaja de las home computer con respecto a las PC es el costo. Mucha gente al no conocer la computación no se anima a hacer una inversión de varios miles de australes, y las home computer les permiten hacer el ensayo con mil o mil quinientos australes.

El usuario no está todavía muy convencido de lo que puede resultar para lo infantil.

El lanzamiento de nuestros sistemas provocó ciertas anécdotas. Por ejemplo, gente que nos decía "no, eso no sirve" o "eso no se puede hacer", o "es imposible que con esa máquina usted pueda realizar lo que promete". Pero lo cierto es que hemos desarrollado sistemas que tienen muy buena acogida, y a la gente que realmente quiere hacer el cambio le resulta de mucha utilidad.

Desde hace un año está apareciendo una nueva mentalidad en el usuario. El hecho de usar la computadora permite aplicarla a cualquier actividad. Desde la actividad comercial e industrial hasta la médica, de archivo de datos, de mailing o procesadora de palabras.

Nosotros estamos en vías de establecer relaciones para la exportación de soft a los países periféricos. Hoy día, la Cámara Argentina de Software está realizando exportaciones con muy buenos resultados, y las perspectivas son muy alentadoras al respecto.

Cuando el público vea la diferencia de lo que es trabajar en las actuales circunstancias, y con un poquito de automatización, como es con la home computer, creo que se abrirá un mercado no tanto comercial como tecnológico que permitirá insertarnos en la computación actual.

En los próximos años todo el mundo va a estar con su pequeña computadora por lo que los costos seguirán bajando. Además, los lenguajes se van simplificando igual que los problemas.

Mucha gente habla de computación y dice "compro la máquina y voy a hacer un curso de

los manuales son tan explícitos que siguiéndoles paso a paso se puede llegar a utilizar la máquina. No se necesita ser programador porque los programas ya están hechos. Todos utilizan una calculadora, pero nadie se pone a estudiar cómo funciona la calculadora. Simplemente hay un librito que indica cómo se deben hacer las operaciones.

"Suplen un vacío"
Héctor Fernández (Pari)

Por supuesto que la actividad de la computación se introduce en las casas a través de los juegos. Luego, cuando el usuario que ya tiene una computadora, descubre que puede [illegible] empieza a sacar las máximas posibilidades. De acuerdo a la profesión del usuario la utilizará para su disciplina. El contador la va a usar para su trabajo, el médico para las historias clínicas, el arquitecto para los diseños y cálculos de estructura y graficación.

Actualmente diversos industriales y comerciantes la están empleando para sus especialidades. Nosotros hemos vendido home computers a estudios contables, estaciones de servicio, farmacias.

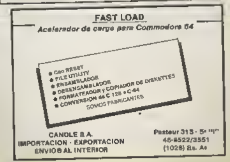
Las computadoras con capacidad profesional o semiprofesional suplen un vacío en la Argentina, porque si esa gente adquiría un equipo PC iba a tener el 60 u 80% de capacidad ociosa por el gasto que había realizado.

La home computer le permite realizar a la pequeña y mediana empresa todo aquello que le está vedado por el alto costo de un equipo PC y sobre todo ven una herramienta más para hacer más eficiente su gestión.

[illegible] es el manejo de los medios de información, y cuanta más información tenga para y de su comercio, más eficiente será. Por ejemplo, las industrias textiles que están utilizando la C-64 hacen un control de stock referido a los colores que se deben fabricar de acuerdo a todo lo vendido, utilizando una consola, una diskettera e impresora de carro ancho. También se utiliza la computadora en forma personal para después de integrar la información a un proceso industrial o comercial mayor.

En el área de la educación, una vez que se decanta la fiebre inicial por los juegos, que indudablemente son excepcionales e iguales a los de video, el padre y mismo el alumno, si va a un colegio que tenga gabinete de computación, se va formando en la disciplina que eligió con mayor eficiencia si utiliza la computación. Lo importante es acercarle el medio a la gente. Algunos lo van a usar, otros no. Pero aquellos que lo hagan, van a obtener un rendimiento más que proporcional que no lo hubieran obtenido sin la computación.

En cuanto a cómo está el mercado, creo que Argentina por su tradicional formación técnica de algunos de sus colegios, está igual de capacitada que muchos países de Latinoamérica como Brasil y México. Con un poco de esfuerzo y persistencia dentro de esta activi-



MERCADO DE LAS HOME

dad que recién se inicia, el país puede ubicarse en los primeros lugares como formadoras de inteligencia y exportadores de productos. Situación que nos puede llevar a obtener divisas para el crecimiento.

Lo veo en los consumidores de computación que van de los 5 a los 18 años, que es la edad en que los chicos se forman. Hay un gran material receptivo de todo lo que sea inteligencia y computación, y como en el resto del mundo, todo ésto se está desarrollando.

Habría que hacer una conjunción entre obtención de divisas vía exportaciones agrícolas y obtención de divisas a través de una actividad que puede hacer que los muchachos se queden trabajando en la Argentina, obteniendo buenas remuneraciones y a la vez exportando. Porque el soft en castellano se puede exportar a Latinoamérica, ya que hay un mercado hispano parlante muy amplio. Pero para eso hay que tener la vocación de que queramos ganar dinero, que se quiera obtener divisas a través de una legislación de fondo que dé apoyo y defienda la propiedad intelectual.

Creo que a través de la resolución 44 puede darse que también se consigan buenos productos exportables en la medida que se combinen con inversiones extranjeras, que unido e lo que podemos llegar a dar de facilidades más la tradicional mano de obra calificada que necesita el hardware y el soft, con los colegios que tenemos, y un correcto apoyo en las universidades; podemos obtener algo que luego se podría trasladar a un desarrollo de la industria privada (que es en definitiva la que consigue mejores salarios). Y también le brindaría un apoyo a toda la estructura estatal que aparentemente es bastante ineficiente. La computación se puede trasladar a los distintos organismos oficiales, y todo integrado provocaría mayor calendad. Las horas libres que le quedaría a la gente se podría aplicar a proyectos que ahora a lo mejor no se hacen producto de que se están llenando papeles o haciendo cosas improductivas.

Omia Informática

Distribuye y vende sistemas de gestión para la C-64 desarrollados por Sistemas Administrativos Modernos S.A. La configuración básica requiere una microcomputadora C-64, una unidad de diskette de 5 1/4, una impresora de 80 columnas y un monitor de video.

Ofrece el **Sistema de Sueldos y Jornales** (SAM-SYJ) que permite llevar la Administración de Personal de una empresa en forma simple y confiable. Sus características principales son las siguientes: amplia flexibilidad para el usuario en el armado de tablas y parámetros del sistema; los campos son denominados en el lenguaje común al área personal; mantenimiento de parámetros, de legajos, de familiares, de conceptos, por categorías; liquidación de un solo legajo, de complementarias, generales, de aguinaldo; aumento de categorías; cálculo de aportes y retenciones; emisión de recibo, libro ley, listado de cambio, totales por código; liquidación de haberes separados por jornalizados, mensualizados, y tres tipos a definir en parámetros para el usuario; mantenimientos de conceptos a liquidar.

Con esta configuración básica se puede manejar hasta 50 legajos y 99 conceptos, los cuales pueden ser depurados parcialmente por el usuario, posibilitando nuevos ingresos.

Además ofrece el **Sistema de Facturación y Stock** (SAM-FAC) que es un sistema totalmente interactivo que permite a una mediana empresa obtener información confiable, planificación y control. Se pueda manejar hasta 600 artículos y 250 clientes.

El **Sistema de Cuentas Corrientes** (SAM-CTA) posibilita llevar las cuentas corrientes en forma rápida y segura facilitando las consultas selectivas por pantalla o impresora. En cada ingreso de Operaciones se informa el saldo del cliente. Se pueden manejar hasta 250 clientes y 1000 operaciones, las cuales pueden ser depuradas por el usuario permitiendo nuevos ingresos.

El **Sistema de Contabilidad General** (SAM-CON) lleva la contabilidad de una empresa pudiendo manejar hasta 350 cuentas y 1300 movimientos, los cuales puedan ser depurados parcialmente por el usuario posibilitando nuevos ingresos.

El **Sistema de Bancos** (SAM-BAN) lleva la cuenta corriente bancaria y la cartera de cheques posdatados de una empresa. Se pueden manejar hasta 100 bancos, 10 códigos de operaciones y 600 operaciones.

Todos los sistemas están diseñados basándose en equipos de mayor envergadura lo que posibilita futuras ampliaciones, de ser necesario, ampliando la configuración básica.

Distribuidora Pari

Trabaja con la línea de Drean Commodore, fundamentalmente la C-64 y la 128. En relación al soft ofrece 1) Programas en castellano de **Sueldos** que emiten recibos, mantienen legajos de personal y liquidan retenciones y aportes patronales con emisión de listado, de **Facturación** en australes, y que trabajan en combinación con el de **Lista de Precios** que también está en australes. 2) Diversos utilitarios con manuales en castellano. 3) En breve lanzará un **Inventario** para 1200 artículos también en castellano.

Distribuye el lápiz óptico "Magic Pencil" con software registrado ley 11723/85

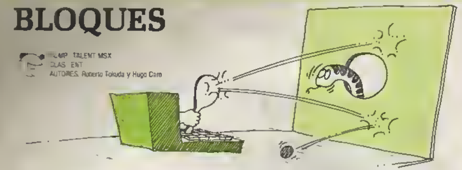
Posee las siguientes características que lo diferencian de lo existente en el mercado: trabaja con efecto spray, se pueden confeccionar texturas propias de fondo de los dibujos a realizar (tiene 36 texturas programadas), trabaja con Fest-Load, tiene cable retráctil, viene con aproximadamente 24 dibujos de muestra de posibilidades de realización, y tiene salida para distintas impresoras.

Además dicta cursos de Logo, Basic, Assembler para C-64, 128 y Z80. También desarrolla cursos especiales para niños hipoacúsicos, dictados por docentes especializados, haciendo realidad las posibilidades de la computación en los diferentes campos de la educación y la salud.

En relación a las posibilidades que ofrece la C-64 y sus periféricos, pone en relieve que los empleados de fabricantes de centrales telefónicas por programa almacenado utilizan la C-64 como apoyo para el diseño de circuitos. También hay programas que convierten a la Commodore en una mini imprenta. Con papel común, en una impresora 803, se pueda confeccionar la papelería propia con impresión de membrete y publicidad. **K64**

GUIA PRACTICA

BLOQUES



AMP: TALENT MSX
CLAS: ENT
AUTORES: Roberto Tokuda y Hugo Caro

El siguiente juego es una combinación de 3 juegos clásicos: El demoledor, el frontón y la víbora que evita ser tocada por algo.
En este caso, se trata de evitar que la pelota que rebota de un borde a otro de la pantalla toque a la calavera. Asimismo, si la pelota toca la cabeza de Bloques (la víbora), también se pierde. Si la cabeza toca la calavera también se pierde.

Para obtener puntos, la pelota debe ir tocando el cuerpo de BLOQUES.
El programa comienza preguntando si utiliza joystick o no. Contestemos "sí" en caso de poseer uno y conectémoslo en el port nº 1. Luego de esto comienza el juego. Para comandar a BLOQUES utilicemos los cursores (o las direcciones del joystick). Si finaliza el juego, pulsemos el disparador (o la espaciadora).

Para jugar nuevamente, o las teclas CTRL+STOP para finalizar.
Es de notar la gran velocidad de reacción del MSX BASIC, ya que la única rutina de código de máquina que se utiliza es para imprimir en la pantalla de alta resolución con gran velocidad. K64

```
1
     BLOQUES
30   DE ROBERTO TOKUDA
40   Y HUGO CARO
50   PARA REVISTA K-64
60 '
0
n MAXFILES=1:CLEAR 100,&HD000:DEFUSR=&HD000:DEFUSR1=&HD010:
  DEFUSR2=&HD021
90 DEFINT A-Z:OPEN"GRP:"FOR OUTPUT AS #1
100 SCREEN 1:CLS
110   CARGA CODIGO MAQUINA E INICIALIZA
120 PRINT:PRINT
130 RESTORE 1430
140 AD=&HD000
150 READ D$:IF D$="FIN"THEN GOTO 170
160 D=VAL("&H"+D$):POKE AD,D:AD=AD+1:GOTO 150
17  INPUT"Utiliza Joystick?(s/n)";I$
15  J=0:TX$="Espaciador":IF I$="S"OR I$="s"THEN J=1:TX$=
    "Disparador"
19  XP=0:YP=0
00 XV=0:YV=0
1  TP=0
22 R=RND(-TIME)
230 SCREEN 2,0,0:KEYOFF
240 ON SPRITE GOSUB 1110
250 PD=4:HIT=0:BL=1:U=USR(0)
260 '
270 ' CARGA SPRITES
```

```
280 '
290 RESTORE
300 SCREEN 2,0
310 FOR K=0 TO 2:SP$="":FOR I=0 TO 7
320 READ R
330 SP$=SP$+CHR$(R)
340 SPRITE$(K)=SP$
350 NEXT:NEXT
360
370   PANTALLA PRINCIPAL
380 '
390 COLOR 15,1,I:CLS
400 LINE(63,0)-(242,186),15,B
410 PRESET(8,24):PRINT#I,"BLOQUES"
420 PRESET(8,56):PRINT#I,"PUNTOS"
430 PRESET(8,64):PRINT#1,STRING$(6.&H9D):U=USR2(0)
440 PRESET(8,96):PRINT#1."TOPE"
450 TP$=RIGHT$("00000"+RIGHT$(STR$(TP).LEN(STR$(TP))-1),6)
460 PRESET(8,104):PRINT#1,USING"&     &";TP$
470 FOR I=20 TO 60 STEP 8
480 XP=65 YP=2 YS=6.YS=6
490 GOSUB 1030
500 XB*100.YB=100
510 PUT SPRITE 1,(XP,YP-2),12 1
520 I=BL:FOR BL=1 TO I
530 GOSUB 1060'NEXT:BL=I
540 '
550 ' PAUSA PARA INICIO
560 '
570 PSET (78,100),0!PRINT #1,"Pulse el "TX$
580 IF STRIG(J)=0 THEN 580
590 LINE (78,100)-(240,108),1,BF
600 BEEP:SPRITE ON
610 '
620 ' RUTINA PRINCIPAL
630 '
640 IF 3=STICK(J) AND 227>XP THEN XP=8+XP:GOTO 680
650 IF 7=STICK(J) AND 71<XP THEN XP=XP-8:GOTO 680
660 IF 1=STICK(J) AND 7<YP THEN YP=YP-8:GOTO 680
670 IF 5=STICK(J) AND 176>YP THEN YP=YP+8 ELSE 720
680 PUT SPRITE1,(XP,YP-2),15,1
690 '
700 ' DIBUJA BLOQUES
710 '
720 LINE (XP,YP)-(6+XP,5+YP),8,BF
730 '
740 ' CHEQUEA REBOTE PELOTA
750 '
760 XB=XB+XS:YB=YB+YS
770 IF 231<XB THEN XS=-ABS(XS):PLAY"L64A":GOTO 790
780 IF 64>XB THEN XS=ABS(XS):PLAY"L64A"
790 IF 1>YB THEN YS=ABS(YS):PLAY"L64A"
800 IF 183<YB THEN YS=-ABS(YS):PLAY"L64A"
810 PUT SPRITE0,(XB,YB),12,0
820 '
830 ' ACERTADO REBOTE
```

```
840
850 IF 8<>POINT(3+XB,3+YB) GOTO 880
860 PLAY"L64B":PAINT (3+XB,3+YB),1:HIT=1+HIT:U=USR1(0)
870 YS=-YS
880 IF 160*BL=HIT THEN BL=1+BL:GOSUB 1060
890 GOTO 640
900
910   RUTINA CHOQUE Y FINAL
920
930 PLAY"L501A":PO=PD-1
940 FOR I=0 TO 1000:NEXT
950 IF PO<>0 THEN FOR I=0 TO 3000:NEXT:GOTO 290
960 IF 10*HIT>TP THEN TP=10*HIT
970 LINE(72,72)-(232,112),1,BF
980 PRESET (120,80):PRINT #1,"FIN JUEGO":PRESET(72,96):PRINT #1,
    "Otro juego:"TX$
990 IF STRIG(J)=-1 THEN 250 ELSE 990
1000 '
1010 ' DECREMENTA CONTADOR BLOQUES
1020 '
1030 LINE(47,168)-(8,160),0,BF
1040 PRINT #1,"BLQ";PO
1050 RETURN
1060 XG=66+INT(RND(1)*22)*8:YG=8+INT(RND(1)*22)*8
1070 PUT SPRITE BL+10,(XG,YG),4,2:RETURN
```

```
1080 '
1090 ' APAGA DETECCION COINCIDENCIAS
1100 '
1110 SPRITE OFF:BEEP
1120 RETURN 930
1130 '
1140 ' DATOS SPRITES
1150 '
1160 DATA &B00111100
1170 DATA &B01111110
1180 DATA &B11111111
1190 DATA &B11111101
1200 DATA &B11111111
1210 DATA &B11111001
1220 DATA &B01110010
1230 DATA &B00111100
1240 '
1250 DATA &B11111111
1260 DATA &B10000001
1270 DATA &B10000001
1280 DATA &B10000001
1290 DATA &B10000001
1300 DATA &B10000001
1310 DATA &B10000001
1320 DATA &B11111111
1330 '
1340 DATA &B01111100
1350 DATA &B11111110
1360 DATA &B10010010
1370 DATA &B10010010
1380 DATA &B11101110
1390 DATA &B01111100
1400 DATA &B01111100
1410 DATA &B01010100
1420 ' DATOS CODIGO DE MAQUINA
1430 DATA 21,51,D0,3E,06,36,00,23
1440 DATA 3D,20,FA,C9,00,FE,41,20
1450 DATA 21,52,D0,06,05,7E,3C,27
1460 DATA E6,0F,77,20,04,23,05,20
1470 DATA F4,21,51,D0,06,06,7E,87
1480 DATA 87,87,C5,E5,21,57,D0,16
1490 DATA 00,5F,19,E5,21,00,08,CB
1500 DATA 20,CB,20,CB,20,48,06,00
1510 DATA 09,54,5D,E1,01,08,00,CD
1520 DATA 5C,00,E1,C1,23,05,20,D6
1530 DATA C9,00,02,00,00,00,00,00
1540 DATA 3C,46,46,46,46,46,3C,00
1550 DATA 1C,3C,7C,1C,1C,1C,7E,00
1560 DATA 3C,46,46,1C,30,60,7E,00
1570 DATA 3C,46,46,0C,46,46,3C,00
1580 DATA 06,0E,16,26,7E,06,06,00
1590 DATA 7C,60,60,7C,06,46,3C,00
1600 DATA 1C,20,40,7C,46,46,3C,00
1610 DATA 7E,46,46,0C,18,18,18,00
1620 DATA 3C,46,46,3C,46,46,3C,00
1630 DATA 3C,46,46,3E,06,06,3C,06
1640 DATA 00,FF,00,FF,00,FF,00,FIN
```

K64

TRUCOS, TRAMPAS Y HALLAZGOS

Capacidades gráficas

Mediante estos dos cortos programas podemos ver como aprovechar las capacidades gráficas del ZX 81 y compatibles. En ambos casos debemos tener la máquina en SLOW.

Una vez que hayan visto los resultados notarán que con pocas líneas y mucha imaginación se puede más que al revés.

```
20 LET A=INT (RND*42)+1
30 LET B=INT (RND*42)+1
40 LET C=INT (RND*2)+1
50 FOR J=1 TO A STEP C
60 PLOT A,J
70 PLOT J,A
80 PLOT B,J
90 PLOT J,B
100 NEXT J
110 GOTO 20
```

```
10 FOR N=40 TO 60
20 FOR I=1 TO N
30 LET R=I*22/N
40 LET T=PI*100*I/N
50 PLOT 32+R*COS T,22+R*SIN T
60 NEXT I
70 PAUSE 100
80 CLS
90 NEXT N
```

Listado particular

Para lograr un efecto visual en un listado, le sugerimos probar el siguiente programa:
10 REM YNC
Luego, tipee:
POKE 16513,56 y presione ENTER,
POKE 16517,147;
ENTER nuevamente.
La computadora debe estar en SLOW.
El problema aparece cuando queremos tomar control de la máquina nuevamente, **dejamos a nuestros lectores la resolución del mismo.**

Sutileza de FAST y SLOW

Una forma menos convencional para pasar al modo FAST, puede lograrse mediante un CALL a la ROM. Concretamente, si hacemos RANDOMIZE USR a la dirección 02E7 (ojo, está en hexadecimal) podremos asegurarnos trabajar en FAST, con la ventaja de que una posterior llamada a 0207 deja a la máquina en el estado en que estuviera antes, sea FAST o SLOW.



PREMIOS DEL MES

COMPUTADORAS - CASSETTES - BECAS

SUSCRIPTORES

GANADOR DEL MES, SORTEO CZ 1000

ALEJANDRO YADRONIK

Villa Celina - Pcia. Bs. As.

FELICITACIONES

DEBERAS RETIRARLA EN NUESTRAS OFICINAS CON LA PRESENTACION DEL D. DE IDENTIDAD

SORTEO ENCUESTA: GANADORES DEL MES

PREMIO 10 BECAS

MARIO CASTINNO
LEANDRO D. BERMERGUI
OSCAR E. MACIA
FERNANDO LUCEMI
JOSE A. BELMONTE
GABRIEL OSTA
GUSTAVO MARTINOH
CARLOS SCHRECK
FERNANDO SOTO
PABLO J. RAI

PREMIO 40 CASSETTES

SILVINA CASTRO KUHAT	Capital
GERMAN CAMUSSI	Córdoba
FERNANDO BONNIM	Dpt. Tucumán
MANGE O DE NADAI	San Martín
LIDIA GANABUAU	Concordia
ADRIAN I. KNANIS	Capital
SILVIA PIRES	Capital
CESAR VENA	Azul
PEDRO A. LESONE	Capital
ALBERTO IRUMBERNI	Rauch
JUAN E. QUIROGA	Córdoba
JUAN CANEOS CEPEDA	Eas Nares
MAURO BETAILLO	Haedo
NAUL JIMENEZ	Capital
ADRIANA MENDIBERNI	Neuquén
RUBEN E. MANSUN	Resistencia
NESTOR A. FERNARA	Río Negro
SANDRA SLIBONA	L. de Zamora
MARCELO A. ZERNIAID	La Plata
MIGUEL A. MONACO	Solano
MARCELO R. BEEVEDERE	Gral. Rodríguez
OSVALDO FAGAIS	Villa Urquiza
JAVIER VAGEIENTE	Córdoba
LUCAS CABANILLAS	Galvez Santa Fe
JUAN A. CASAELLA FERNANDO	La Plata
CLAUDIA S. ORTIZ	Mercedes
EMILIO ANARAKI	R. de Escalada
FEDERICO PANDDI	Victoria
SERGIO E. PEARIN	Ionia Alta
MARIA S. TOSCANO	Azul
EDUARDO BENGUÑO	Ramos Mejía
LUCAS F. VILLAGRA	Tucumán
LEANDRO JESUI	Paraná
EDGAR A. BEA	San Luis
MARTIN GALIGN	Marcos Juárez
ORLANDO O. CORONEE	Caseros
MIGUEL A. NIUS	Rawson
FERNANDO QUEVEDO	Villa María
JUAN A. AFTER	Cornelles
EZEQUIEL FNESINI	Río Gallegos
ESTEBAN EIF NEVENNY	La Plata

Estos premios podrán retirarse en la Administración de K64 Cerrito 1320 1º P. Capital con documento de identidad en el horario de 10 a 12 y 15 a 17 hs. Quienes viven en el interior del país, pueden solicitar que se les remitan los premios por correo

RAID OVER MOSCOW



CLASIFICACION: JUEGO DE ACCION
COMPUTADORA: COMMODORE 64
CONFIGURACION: 64 K
FACTON K-64: 9
HECHIZO: 9
DOCUMENTACION: N/D

Esta vez nos loca astar al frante de una misión para detener y destruir los misiles nucleares que la Unión Soviética ha lanzado contra distintos puntos del mundo libre.
La acción de RAID OVER MOSCOW varía en su tipo y nivel de dificultad a medida que avanzamos en el juego. La acción comienza con una vista del hemisferio norte dal planeta, mostrando la detección del lanzamiento de misiles de uno da cuatro silos situados an Rusia. En el nivel de principlantes, tenemos sólo siete minutos para detener la guarra nuclear. Como piloto de caza que somos, debemos rápidamente despegar, ponernos en camino a Rusia y atacar los silos da misiles.
Ten solo para despegar se necesita una habilidad considerable, dado que debemos manejarnos en un ambiante de ascasa gravedad y, maniobrar la nave en estas condiciones, no es cosa fácil.
Los controles son muy realistas. El secreto para podar daspegar con éxito consiste en tomar control da la nave apanas ésta se lavanta del suelo y al estar a la altura correcta abrír las puertas del hangar.
Se pueden volar hasta nueve aviones por juego y es una pena que al comienzo se pierdan la mayoría de ellos tratando de salir del hangar, siando ésta parte la más aburrida del juego.
El territorio enamigo setá defandido por misiles guiados por calor que se aproximan por la retaguardia y numerosas armas de tierra. Se deba volar a baja altura para evitar los misiles, a la vez que se avitan los árboles y se destruyen las estaciones de tierra. La respuesta del avión as eficienta y al scroll dal terrano esta muy bien logrado.
Destruir los silos de los misiles no es una empresa tan difícil. Estos están defendidos por cohetas y la aviación enemiga. Sólo el silo dal madio as el importanta, pero los de los coslados nos dan más puntos y bonus. La paqueña vantana por la que disparan es su punto débil. Una vez que estamos alineados con la vantana, ésta se vuelve azul. Destruido un sitio de lanzamiento se pasa al próximo, hasta que se dastruyan tres.
Luego pasamos al ataque del Centro de Defensa Soviético; es, sin dudas, una de las pantallas mas divertidas. Tenemos que abrirnos paso entre soldados enemigos y un tanque. Una vez hecho esto llegamos al cuarto del reactor. Esta as la parte final dal juego y sólo les diremos que no será nada fácil salir con vida de esta lugar.
Los gráficos del juego están muy bian dafinidos, y el sonido as sumamente realista. La nave responde a los controles an forma rápida y eficiente, y el juego tiene un nival de dificultad realmante alto.
Pese a tener tres niveles de dificultad, incluso an al primaro, se haca difícil llegar a las partes finales del juego (an aspecial las dos últimas) que son las más divertidas.
RAID OVER MOSCOW es un juego bien hecho, que no nos va aburrir por el juego en sí mismo, aunqua nos pueda decepcionar por no poder lograr una buena preformance hasta haber pasado unas cuantas horas con él.

ANT ATTACK



CLASIFICACION: JUEGO DE ACCION
COMPUTADONA: SPECTRUM · TK-90X
CONFIGUNACION: 48 K
FACTOR K 64: 7
NECHIZO: 7
DOCUMENTACION: N/D

Nos ancontramos sin dudas frente a un clásico de la linea SPECTRUM Esta juego tiene como principal característica que no desciende de ningún otro clásico (por aj. derribar invasores del espacio) y utiliza una nueve técnica tridimensional que su autor, Sandy White, conserva como su más preciado secreto.

El juego comlanza con una presentación en donde se nos introduce en la misión que debemos llevar a cabo y, algo no muy común, somos requeridos ecerca de nuestro sexo. Parece que por lin les diaron un lugar a las chicas en los juegos donde el heroe rescata a la doncella, porque aca la situación puede ser Invertida y ser la doncella, porque ace le situación puede ser invertida y ser la doncella quien rescata al heroe.

Una vez dentro de la ciudad de las hormigas debemos rescatar a quian se halla prisionera y sacarla de la ciudad. Esto debe hacerse antes de que el reloj llague a cero, en cuyo caso le aventure habrá concluido, y sin que las hormigas nos piquen mas de veinte veces, tanto a nosotros como a la doncella que corre a nuestras espaldas.

La ciudad está muy bien realizada y aqui es donde se encuentra la mayor etracción del juego, en el electo tridimensional. Es como si nosotros vieramos la acción a través de una de cuatro cámaras de televisión que se hallan en las esquinas de una habitación. Mediante cuatro leclas podemos seleccionar une de las cámaras y asl poder apreciar la acción desde distintos angulos. De este modo nunca quedaremos ocultos por un edificio, basta con cambiar el ángulo de visión.

El control del movimlento debe hacerse mediante el teclado, dado que en su versión original no incluye la opción de utilizar joystick. Para matar a las hormigas disponemos de veinta bombas que puadan sar errojadas más o menos lejos según la tecla que apretemos. Da todos modos el funcionamiento de estas granadas no es del todo efectivo y cuando más las necesitamos misteriosamente no hacen blanco en las hormigas.

Los movimlantos están muy bian logrados y el scroll de la pantalla, a medida que nos movemos por la ciudad, as suave y cómodo para la visión.

Los gráficos son buanos, no tienen gran colorido porque no es realmente necesario y el juego posee escasos sonidos.

El programa está grabado a velocidad normal, y no tuvimos problemas para su carga.

Anl Attack es un juego que tardara mucho an resultarnos aburridos y es uno de los clásicos que no deban laltar an una buena biblioteca de juegos

Intercambio TK 2000

Quiero comunicarme con todos aquellos que posean una TK 2000 y estén interesados en intercambio de software e información sobre dicho microcomputador.
Desgraciadamente en nuestro medio nadie se ha interesado en ocuparse del mismo, y toda la bibliografía y programas vienen del Brasil.
Por eso intento reunir a quienes se interesen, utilizando la revista de mayor difusión en Argentina.
Desde ya muchas gracias.

Hugo H. Campanelli
Suárez 1729 8 P. "C"
(1288) Capital

TI en Búsqueda:

Me gustaría comunicarme con Clubes de Usuarios de TI99/4A

Adriana Mendiberri
Eduardo Talsro 975
8300 - Neuquén

En esta sección atendemos todas aquellas consultas y sugerencias que nuestros lectores deseen realizar. Para ello sólo debe dirigirse a esta redacción, sección "Consultas".

K64

Publicamos tu dirección completa con la seguridad de que se pondrán en comunicación contigo.

34 Columnas

En la revista de Diciembre, en la sección "Trucos Trampas y Hallazgos" dan un programita para escribir en 34 columnas con la TS 1000, pero borra las marcas de fin de línea y al hacer PRINT o CLS se pierde el control.
La solución está en estos dos programitas. El primero es similar al que publicaron ustedes pero deja el archivo de pantalla con sus marcas de fin de línea tal como estaba antes.
El segundo permite imprimir cualquier mensaje en la pantalla de 34 columnas. Está compuesto por 3 subrutinas y es muy sencillo, para pasar a la modalidad de 34 columnas hacer: GOSUB 9600; para imprimir el mensaje GOSUB 9000, habiendo colocado el mensaje en la variable a$ de la siguiente forma: los dos primeros caracteres son el número de línea (1 a 22), los dos restantes son los de la columna (1 a 34)
EJEMPLO:
Para hacer PRINT AT 15, 7; "HOLA", DEBE-MOS HACER: LET A$ = "1406 HOLA" Y GOSUB 9000
Para volver a modalidad normal: GOSUB 9500
También les ofrezco mi aporte a los crash de lujo; prueben con PRINT USR 2.

2068 ó TK 90X

Les mando estas líneas, primero, para felicitarlos por su exitosa revista.
La otra razón es preguntarles si los programas que son para tipear, hechos con una TS 2068, pueden tipearse con una TK 90X, ya que lo intente con el programa GATE y no pude hacerlo puesto que el paso 7130 SOUND 8,0,... no tiene equivalente en mi máquina.

Guillermo F. Rutman
Rosario

K64

Con respecto a su pregunta, la compatibilidad entre una 2068 y la TK 90 es la misma que entre la 2068 y la SPECTRUM. Por lo tanto no hay muchas sentencias que no

Listado 1

```
10 LET F=PEEK 16396+256*PEEK 16397+1
20 FOR A=1 TO 22
30 FOR B=1 TO 34
40 POKE P,1
50 LET F=P+1
60 NE  B
70 NE  A
100 IF INKEY$="" THEN GOTO 100
200 LET Z=PEEK 16396+256*PEEK 16397
210 FOR X=Z TO Z+726 STEP 33
220 POKE X,118
230 NEXT X
```

Listado 2

```
100 GOSUB 9
120 LET A$= . 01ESTA ES UNA LINEA DE 34 CARACTERES
130 GOSUB 
500 IF INKE $="" THEN GOTO 500
510 GOSUB 95 0
520 STOP
9000 REM IMPRIMIR EN 34 COL.
9010 LET  =Z+ VAL A$( TO 2)*34+VAL A$(3 TO 4)) 35
9020 FOR A=1 TO LEN A$ 4
9030 POKE X+A,CODE A$(A+4)
9050 RETURN
9500 REM  VOLVER A LA NORMALIDAD
9510 FOR  =Z+33 TO Z+726 STEP 333
9520 POKE  ,118
9530 NEXT Y
9540 RETURN
9600 REM PREPARAR ARCH DE IMAGEN
9610 LET Z= PEE  16396+256*PEEK 16397
9620 FOR Y=Z+33 TO Z 726 STEP 33
9630 POKE Y,0
9640 NEXT
9650 RETURN
```

tengan equivalente, y una de alias es el comando SOUND.
Si bien no es lo mismo, éste puede ser reemplazado con el comando SOUND de la TK 90, aunque su frecuencia y duración deberá ser probada hasta dar con sonidos de tu gusto. Lo que ocurre es que la TS2068 permite en ese comando, mayor control sobre el sonido generado; por eso lleva un argumento de más de dos números.

Sonido con TK 85

Ante todo quiero felicitarlos por la revista, esté super buena.
Soy poseedor de une TK 85 y quisiera saber dos cosas.
1 - Me he enterado que existe un generador de sonido para la TK 85. ¿Cómo se usa y donde se lo conecta a la computadora?
2 - Si ustedes o algún lector me pudiera enviar una copia del programa "Simulador de vuelo", ya que acá se agotó rápido la revista y no me fue posible adquirirla.
Desde ya muchas gracias

Aníbal Adrián Londero
Don Bosco 1660
(3100) Paraná (E.R.)

K64

1 - El generador de sonido que conocemos se conecta al port de expansión trasero de la máquina y para su uso deben utilizar pokes adecuados, aunque existe software que facilita esta operación.
2 - Esperamos contestar a tu llamado.

TI vs. Commodore

Soy estudiante de segundo año y estoy próximo a comprarme un microcomputador TI 99/4A. Ustedes el responder a un lector le dijeron que era un modelo enticuado y que la más conveniente era la Commodore.
Yo, por un lado veo al sistema operativo de la TI 99 más "fácil", comparado al de COMMODORE. Quiero que me informen si estoy en buen camino y si para ustedes ser anticuado es ser ineficiente. En todo caso me podían decir los pro y los contra de estas dos máquinas.

Sergio Leguizamón
Wilde

K64

En primer lugar, el que la hayas decidido a comprar una microcomputadora ya es un buen camino.
En cuanto a qué máquina elegir, si la TI o la COMMODORE, no podemos decirte que compres una u otra. Esta decisión te toca tomarla a vos. Analizar los pro y contras de cada máquina nos llevaría hojas, pero podemos darte un par de bases sobre las cuales fundamentar tu decisión.
1 - Memoria libre para el usuario: la C-64 tiene 64 K Ram de fábrica, la TI tiene muchos menos. Averigue cuanto te costaría expandirla.
2 - Expansiones: ¿Cuánto cuesta ponerle diskettera a una u otra máquina? Aunque no lo pienses comprar de inmediato, es un factor importante si pretendes darle algún uso serio a la computadora.
3 - Software: qué cantidad, variedad y precio. Comparalos.

TRUCOS Y TRAMPAS

Debo felicitarlos por la revista y decirles que les sección "Trucos y Hallazgos" es realmente interesante ya que poseo una CZ1500 y tengo un programa (Frogger) imposible de breakear y siguiendo sus consejos pude hacerlo y desarmarlo todo.
¿Qué dice en la línea 200 de Interceptor- Galáctico?

Eduardo Dorola
S.A de Padua

K64

Gracias, y por lo de la la línea 200, ya salió en el número anterior la rectificación.

Zeus

Les escribo con motivo de hacerles la siguiente consulta: tengo una 2068 y me ha resultado de gran interés el artículo "Computadoras que hablan".
Con él se ofrece un programa en código máquina. Me gustaría saber cómo tipear este programa y si se necesita un ensamblador. En tal caso, podrían informarme ¿qué es, como es y cómo se utiliza un ensamblador?

José Luis Coccoz
Venado Tuerto - Santa Fe

K64

El programa a que haces referencia debe ser cargado con un ensamblador. En cuanto al uso del mismo podrás verlo en forma detallada en la nota "CARGANDO COMO LOS DIOSES" donde se explica cómo utilizar el ZEUS ASSEMBLER.



Bolsa de Usados

Compro impresora Omnis 850 para Texas TI 99/4A.
Héctor M BEIM
Cerrito 159
ROSARIO
TE. 820188

Vendo TI 99/4A con 5 módulos (Basic extendido, Parsec, Microcirugía, Ajedrez; Home financial Dec) y 105 programas (incluye curso Basic), cable p/grabador; funda, manuales; etc. Todo original y en perfecto estado: ₳ 350.-

TE. 797-5334 - Sergio
Capital Federal

Vendo pantalla AMDEK 12" Video Monitor Model 310/310A W/CARD sin uso: ₳ 350.-

TE. 312-2141/8505 de Lun. a Vier. de 10 a 19 hs.

• VENDO TK-85 en perfecto estado, con joystick y 20 juegos. Todo ₳ 140.- Christian Scivetti, Sarmiento 4995, esquina Rio Gallegos, Ezpeleta, Pcia de Bs. As.

• VENDO TK-83 en expensor, cables, fuente, manual y caja original, más 10 programas, todo como nuevo Guillermo Olmos, San Martín 58, (7240) Lobos, Pcia. de Bs. As.

• VENDO Interfase 1; microdrive; 4 cartuchos, software e impresora térmica Alphacom 32, Te.: 825-2302

• VENDO Cartridge emulador Spectrum para TS 2068, Te.: 825-2302





SORTEO-ENCUESTA K64

LLENE ESTE CUPON Y PARTICIPE DEL SORTEO MENSUAL

50 PREMIOS:
40 CASSETTES Y 10 BECAS PARA CURSOS BASIC

NOMBRE: .... EDAD: ...... OCUPACION: ...... TEL: ....

DIRECCION: .... C.P.: ...... LOCALIDAD: ...... PCIA.: ....

ENCUESTA

COMPUTADORA: ☐ CZ 1000 ☐ CZ 1500 ☐ CZ 2000 ☐ TIMEX 2068 ☐ TK 83 ☐ TK 85 ☐ TK 90
☐ C-16 ☐ C-64 ☐ TI 99/4A ☐ Otras .... ☐ NO TENGO AUN

ME GUSTARIA VER:

MAS	IGUAL	MENOS	
☐	☐	☐	PROGRAMAS DE APLICACION ESPECIFICA
☐	☐	☐	PROGRAMAS EN BASIC
☐	☐	☐	PROGRAMAS EN LOGO
☐	☐	☐	PROGRAMAS EN LENGUAJE DE MAQUINA
☐	☐	☐	PROGRAMAS EN OTROS LENGUAJES
☐	☐	☐	ANALISIS DETALLADOS DE LOS PROGRAMAS
☐	☐	☐	NOTAS PARA BEGGINERS

MAS	IGUAL	MENOS	
☐	☐	☐	JUEGOS
☐	☐	☐	CALIFICACION DESCRIPTIVA DE PROGRAMAS DE JUEGO
☐	☐	☐	PROGRAMAS DE APLICACIONES COMERCIALES
☐	☐	☐	PROGRAMAS EDUCATIVOS
☐	☐	☐	HARDWARE

QUE ES LO QUE MAS TE GUSTA DE K64?

QUE ES LO QUE MENOS TE GUSTA?

Envíarlo a: K64 Computación Para Todos - Cerrito 1320 1° (1010) Buenos Aires, Rep. Argentina



REVISTA PARA USUARIOS DE
Drean commodore
Nº 3 A 2.00
REP. ARGENTINA
Programas
DIRECTORIO SIMULTANEO
MEZCLADOR DE PALABRAS
CALCULOS MATEMATICOS
PROGRAMACION ESTRUCTURADA
Assembler:
Direcciones de memoria

¡YA!
CLUB DE USUARIOS
Talent MSX
Chile 1345



El Club de Usuarios de MSX ya funciona en Chile 1345

Invitamos a los felices usuerios de la TALENT MSX el curso gratuito da introducción el fabuloso mundo de MSX.

Perticipe del Club de Usuarios de MSX y encuéntrese con sus amigos que también tienen le TALENT MSX, e intarcambiará programas, detos y chimentos.

Podrá probar todos los eccesorios de la línee MSX, ¡¡desde disketteras hasta robots!!

Podrá var y leer todo lo que la interese sobre le norma MSX: catálogos, libros y revistes de todo el mundo. Todo con le seguridad, respaldo y seriedad que sólo TALENT puede brindarle.

¡Para inscribirse, no olvide traer su factura de compra!

Club Talent MSX